



VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Timo Vilpunaho

KELAINSOVELLUKSEN JA KÄYTTÖ- LIITTYMÄN OHJELMOINTI

Tekniikka ja liikenne
2014

TIIVISTELMÄ

Tekijä	Timo Vilpunaho
Opinnäytetyön nimi	Kelainsovelluksen ja käyttöliittymän ohjelmointi
Vuosi	2014
Kieli	suomi
Sivumäärä	78 + 3 liite
Ohjaaja	Juha Nieminen

Training centerin alkuperäistä kelainta ohjattiin taajuusmuuttajilla ja se haluttiin päivittää vastaamaan paremmin teollisuuden prosessia, missä ohjauksesta vastaa logiikka. Prosessin hallintaa varten haluttiin käyttöliittymä, jota voitaisiin operoida kosketusnäytöltä.

Kelainsovelluksen keskeisin ominaisuus on nauhan kireyssäädön tarkkuus kelauksen aikana. Kireys saatiin aikaiseksi laskemalla moottorille momenttiohje, joka koostui useasta osasta. Näiden momenttiohjeen osien laskentatarkkuus ja toimilaitteiden kyky toteuttaa pyydetty momentti, määritteli toteutuneen nauhankireyden tarkkuuden. Logiikan ja taajuusmuuttajien välinen väyläliikennöinti haluttiin toteuttaa kahdella eri väylällä, joista käytettävän väylän pystyi valitsemaan käyttöliittymästä.

Työn lopputuloksena saatiin tilauksen mukainen laite. Kelainta pystyi hallitsemaan kosketusnäytöllisen käyttöliittymän välityksellä ja siihen tuli myös muita tilaajan haluamia lisäominaisuuksia.

ABSTRACT

Author	Timo Vilpunaho
Title	Programming of a Winder and HMI
Year	2014
Language	Finnish
Pages	78 + 3 Appendices
Name of Supervisor	Juha Nieminen

The objective of the thesis was to program a winder and an HMI for it. The original winder was controlled with a custom made drive application. The customer wanted to update the winder to correspond more with real process where drives are controlled by PLC. The customer also wanted to add a HMI to control the process.

The main purpose of the winder is to keep web tension steady in the set point while winding. The web tension is controlled with the motor torque reference. This torque reference consists of many parts. The accuracy of the web tension accuracy depends on the accuracy of the calculations made to form the final torque reference. Communication between drives and the PLC was implemented with two different field buses. The operator is able to choose the fieldbus used from the HMI.

As a result, the winder worked as the customer wanted. Actuators were controlled with the PLC via two optional fieldbuses which can be chosen from the HMI. From the HMI the operator is also able to control the process.

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	9
2	KELAINDEMOLAITE.....	10
3	KÄYTETYT LAITTEET.....	12
3.1	Moottorit.....	13
3.2	Taajuusmuuttajat.....	13
3.3	Logiikka.....	14
3.4	HMI-PC.....	14
3.5	Kireysanturit ja vahvistin.....	14
4	KÄYTETYT OHJELMISTOT.....	15
4.1	TIA-Portal.....	15
4.2	InTouch.....	15
4.2.1	SMC Orchestra system management console.....	15
4.2.2	InTouch Window Maker.....	16
4.2.3	Window Viewer.....	16
4.3	NCDrive ja Live.....	16
5	TAAJUUSMUUTTAJIEN PARAMETROINTI.....	17
5.1	PPO17.....	
5.2	Prosessidatan määrittely.....	18
5.3	CW ja SW.....	19
6	LOGIIKAN OHJELMOINTI.....	23
6.1	Hardware-asetukset.....	23
6.2	Taajuusmuuttajan tyyppilohko.....	29
6.3	Laskentalohkojen ohjelmointi.....	30
6.3.1	Voiman mittaus.....	30
6.3.2	Nopeuden mittaus.....	32
6.3.3	Lämpötilojen mittaus.....	32
6.3.4	Halkaisijan laskenta.....	33
6.3.5	Nauhan pituuden laskenta.....	34
6.3.6	Pituusennakon laskenta.....	35

6.3.7	Pysäytysten aikaisten kierrosten laskeminen	36
6.3.8	Kelan massan laskenta	38
6.3.9	Derivointi	39
6.3.10	Massahitausmomentin laskenta.....	39
6.4	Kelausohjelma.....	40
6.4.1	Kiristys	41
6.4.2	Nopeusohje.....	42
6.4.3	Momenttiohjeen muodostus	44
6.4.4	Kelaimen pysäytys	49
6.4.5	Loop-toiminto	50
6.4.6	Kelauksen aikaiset rajoitukset.....	50
6.5	Kiristyssekvenssi.....	51
6.6	Ryömintä.....	52
6.7	Kenttäväylän valinta	53
6.8	Vika- ja varoitussanat	54
6.9	Prosessi- ja taajuusmuuttajien tilatieto.....	58
6.10	Lämpötilan säätö	58
6.11	Jarrujen ohjaus	60
7	KÄYTTÖLIITTYMÄ	61
7.1	Yhteyden muodostaminen	61
7.2	Käyttöliittymän ohjelmointi.....	63
7.2.1	Pääsivu	63
7.2.2	Monitorointi-ikkuna	68
7.2.3	Huoltoikkuna.....	69
7.2.4	Window Viewer-asetukset	70
7.2.5	PC:n asetukset	70
8	ESIMERKKIAJO	72
9	KÄYTÖN RAJOITUKSET.....	76
10	YHTEENVETO JA POHDINTA.....	77
	LÄHTEET.....	78
	LIITTEET	

KUVIO- JA TAULUKKOLUETTELO

Kuvio 1 Kelain	s.10
Kuvio 2 Järjestelmäkaavio ja verkkorakenne	s.12
Kuvio 3 PROFIBUS-väylän PPO-tyypit	s.17
Kuvio 4 Taajuusmuuttajan neljä prosessidatan määrittelyt	s.18
Kuvio 5 Bypass-PROFIDRIVE CW	s.20
Kuvio 6 CW-bit12 määrittely	s.21
Kuvio 7 SW Bypass-PROFIDRIVE-tilassa	s.21
Kuvio 8 SW-bittien 11 ja 12 määrittely	s.22
Kuvio 9 Logiikan komponentit	s.23
Kuvio 10 Ethernet -asetukset	s.24
Kuvio 11 PROFIBUS-liikennöintiasetukset	s.25
Kuvio 12 GSD- ja GSDML-tiedostot	s.25
Kuvio 13 Toimilaitelista	s.26
Kuvio 14 PROFIBUS slave-asetukset	s.27
Kuvio 15 PROFINET I/O PPO-tyypit	s.27
Kuvio 16 Taajuusmuuttajan PROFINET I/O-asetukset	s.28
Kuvio 17 Logiikan verkkoasetukset	s.29
Kuvio 18 Taajuusmuuttaja tyyppilohko	s.30
Kuvio 19 Voiman mittauksen skaalaus	s.31
Kuvio 20 Väkipyörän voimat	s.31
Kuvio 21 Nauhan nopeuden laskentalohkon ohjelmakoodi	s.32
Kuvio 22 Lämpötilan mittauslohko	s.33
Kuvio 23 Halkaisijan laskentalohkon ohjelmakoodi	s.33
Kuvio 24 Pituuden laskentalohkon ohjelmakoodi	s.35

Kuvio 25 S-rampin viive	s.36
Kuvio 26 S-käyrän ja suoran vertailu	s.37
Kuvio 27 Pituusennakkolohkon ohjelmakoodi	s.37
Kuvio 28 Taajuusmuuttajan nopeusohje ja toteutunut nopeus	s.38
Kuvio 29 Massan laskentalohkon ohjelmakoodi	s.39
Kuvio 30 Derivointilohkon ohjelmakoodi	s.39
Kuvio 31 Massahitausmomenttilohkon ohjelmakoodi	s.40
Kuvio 32 GRAPH-editorilla toteutettu kelaussekvenssi	s.41
Kuvio 33 Alkukiristuksen ohjelmakoodi	s.42
Kuvio 34 Ramppilohko	s.43
Kuvio 35 Taajuusmuuttajan saama nopeusohje ja toteutunut nopeus	s.44
Kuvio 36 Momenttiohjeen ensimmäisen osan ohjelmakoodi	s.44
Kuvio 37 Kitkan kompensoinnin ohjelmakoodia	s.46
Kuvio 38 Linjamoottorin nopeusvirhe	s.47
Kuvio 39 Continues controller-lohko	s.48
Kuvio 40 Kelauksen aikainen momenttiohje ja toteutunut momentti	s.49
Kuvio 41 Kiristyslohko	s.52
Kuvio 42 Lähtöosoitteen valintalohko	s.53
Kuvio 43 Tulo-osoitteen valintalohko	s.54
Kuvio 44 Taajuusmuuttajan vikasanan selitykset	s.55
Kuvio 45 Vikasana-DB	s.56
Kuvio 46 Vikasanan tulkintalohkon ohjelmakoodia	s.57
Kuvio 47 Vikasanan tulkintalohkon ohjelmakoodia	s.58
Kuvio 48 Tuulettimien nopeusohje	s.59
Kuvio 49 SMC-liikennöintiasetukset	s.62

Kuvio 50 Käyttöliittymän pääsivu	s.64
Kuvio 51 Vikatiedon linkitys integer-muuttujaan	s.65
Kuvio 52 Yhteyden tarkistus scripti	s.65
Kuvio 53 Window Maker security-valikko	s.66
Kuvio 54 Acceslevel-toiminnon liittäminen objektiin	s.67
Kuvio 55 Sisäänkirjautumisikkunan avaus scripti	s.67
Kuvio 56 Monitorointi-ikkuna	s.68
Kuvio 57 Historiaikkuna	s.68
Kuvio 58 Maintenance-ikkuna	s.69
Kuvio 59 Ohjeen avaus scripti	s.70
Kuvio 60 Kireyshistoria	s.72
Kuvio 61 Aukikelaimen mittaukset	s.73
Kuvio 62 Nopeusvirhe	s.74
Kuvio 63 Momenttiohje ja toteutunut momentti	s.75
Taulukko 1 Moottoreiden kilpiarvot	s.13
Taulukko 2 Logiikan komponentit	s.14

1 JOHDANTO

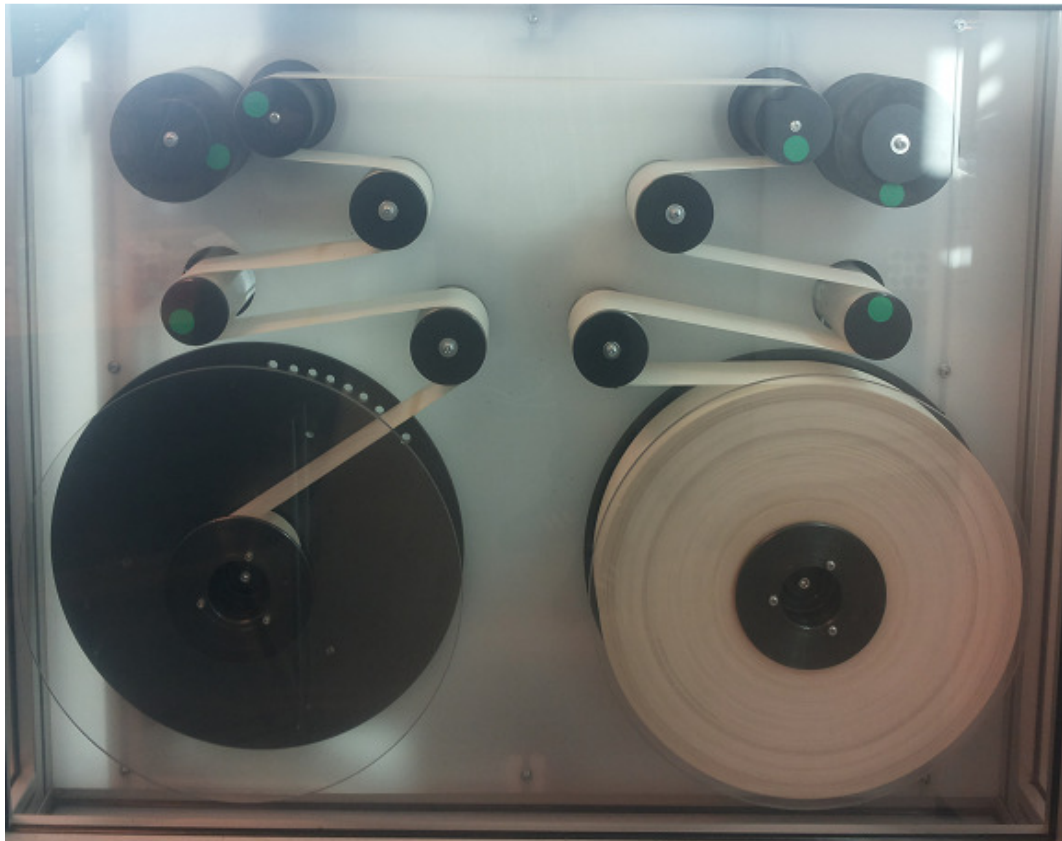
Opinnäytetyön tilaajana oli Vacon Oyj, joka on markkinoiden johtava pelkästään taajuusmuuttajien valmistukseen keskittynyt yritys. Vacon työllistää maailmanlaajuisesti noin 1600 ihmistä. Opinnäytetyön tarkoituksena oli uudistaa Vacon Training Centerin harjoitus- ja esittelykäyttöön valmistettu kelain. Alkuperäisen kelaimen ohjaus oli toteutettu Vaconin NXP-taajuusmuuttajille tehdyllä erikoissovelluksella.

Kelaimen uudistaminen koostui kahdesta eri opinnäytetyöstä. Ensimmäinen työ oli johdotusten uusiminen, taajuusmuuttajien kontrolliosien päivitys ja taajuusmuuttajien ohjelmistojen päivitys ja parametointi. Työn ensimmäisen osan suoritti aikuisopiskelija Jani Oravasaari.

Toinen osa oli logiikan ja käyttöliittymän ohjelmointi. Työn tavoitteena oli saada laite kelaamaan nauha automaattisesti aukikelaimelta kiinnikelaimelle ja tehdä laitteelle käyttöliittymä, josta prosessia voitiin hallita. Työn edetessä ohjelmaan lisättiin tilaajan toivomia lisäominaisuuksia.

2 KELAINDEMOLAITE

Kelaimen pääkomponentit ovat kuvion 1 alaosassa sijaitsevat aukikelain, kiinnikelain ja yläosassa olevat telat, joiden välistä lasikuitunauha kulkee.



Kuvio 1 Kelain

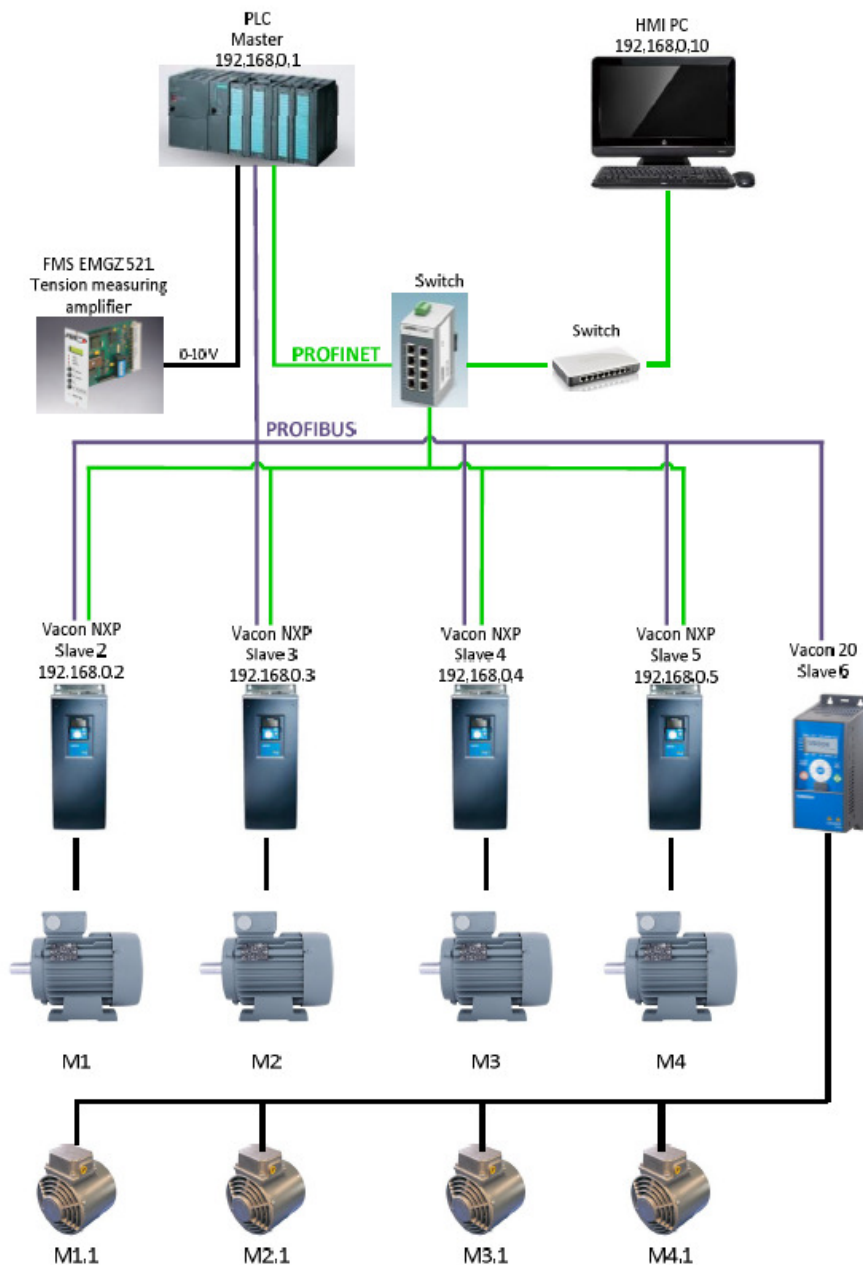
Kelaimen tarkoituksena on kelata lasikuitunauha aukikelaimelta kiinnikelaimelle pitäen nauhan kireys mahdollisimman tasaisesti asetusarvossa. Nipit ovat kiristetty niin tiukkaan, ettei lasikuitunauha pääse luistamaan. Tämän vuoksi laitteeseen muodostuu kolme eri ratakiireyttä. Näistä ratakiireyksistä kelainten ja linjateloiden välistä kireyttä mitataan ja säädetään. Linjateloiden välistä kireyttä ei mitata, eikä pyritä säätämään. Kelauksen aikana aukirullainta ohjataan momenttisäädöllä ja annetulla momenttiohjeella saaden aikaiseksi halutun ratakiireyden. Linjatelat toimivat nopeussäädöllä ja niillä ohjataan nauhan nopeutta. Kiinnikelaimen yläpuolella oleva linjatela toimii pääkäyttönä linjassa. Toisella linjatelalla on pienempi nopeusohje ja pieni momenttiraja. Sen tarkoituksena on pitää linjan nauha kireänä.

Kiinnikelain toimii momenttiohjauksella moottorina ja pyrkii aukikelaimen tavoin pitämään asetetun ratakireyden.

Laitteen suurimmaksi ratanopeudeksi on asetettu 300 m/min ja suurimmaksi kireysohjeksi 50 N. Nauhaan kohdistuva voima mitataan vahvistimeen kytketyillä pyörivillä kireysantureilla. Logiikka saa kireystiedon vahvistimelta jänniteviestinä. Nopeus mitataan moottoreissa olevilla takometreillä. Takometrit on kytketty taajuusmuuttajien encoder-optiokorttiin, jotka muuttavat takometrin lähettämän tiedon nopeustiedoksi.

3 KÄYTETYT LAITTEET

Kelaimessa käytettyjä laitteita ovat moottorit, moottorituulettimet, taajuusmuuttajat, logiikka, HMI-PC, kireysanturit ja mittavahvistin. Logiikan I/O:hon kytkettiin kelaimeen kyljessä olevat painikkeet ja indikointivalot. Myös rajatiedot tuotiin I/O:hon. Kuviossa 2 on esitetty kelaimeen pääkomponentit ja verkkorakenne.



Kuvio 2 Järjestelmäkaavio ja verkkorakenne

3.1 Moottorit

Kelaimessa on neljä identtistä VEM:in kolmivaiheoikosulkumoottoria, joita ohjattiin Vaconin NXP-taajuusmuuttajilla. Moottoreiden kilpiarvot ovat taulukossa 1. Kelat ja linjatelat oli kytkettynä suoraan moottorin akseleihin. Moottoreiden rungon lämpötilaa mitattiin PT-100 elementeillä ja moottoreiden jäähdytys toteutettiin erillisillä tuulettimilla.

Taulukko 1 Moottoreiden kilpiarvot

U	230/400 V
Cosφ	0,69/0,63
P	1,1 kW
I	5,5/3,15 A
Nn	925–940 rpm

3.2 Taajuusmuuttajat

Kelainlaitteen moottoreita ohjattiin Vaconin ilmajäähdytteisillä NXP-sarjan taajuusmuuttajilla. NXP:lle on tarjolla useita eri sovelluksia. Opinnäytetyössä käytettiin NX-sarjalle tehtyä APFIF40 System interface application II:sta. Ohjelma tarjoaa paljon parametreja oikosulkumoottorin ja kestopagneettimoottorin säätöön.

Logiikan ja taajuusmuuttajan välisen liikennöinnin kannalta käytetty ohjelmisto ei tuonut etuja, mutta laitteiston käyttöönotossa siitä oli hyötyä. NXP:n parametroida toteutettiin Vaconin omalla ohjelmalla, NCDrivellä.

Kaikkiin taajuusmuuttajiin kytkettiin seuraavat optiokortit: vakio optiokortti OPT-A1, encoderkortti OPT-A5, PRPFIBUS-DP OPT-C3 ja PROFINET I/O OPT-CP. Kahteen taajuusmuuttajaan lisättiin myös PT-100-optiokortti OPT-B8.

Moottoreiden erillisiä tuulettimia syötettiin yhdellä Vacon 20-taajuusmuuttajalla. Moottorit kytkettiin rinnan taajuusmuuttajaan. Vacon 20-taajuusmuuttajan parametroida tehtiin Vacon live -ohjelmalla.

3.3 Logiikka

Logiikkana oli Siemensin S7-300. Logiikan CPU on 315-2 PN/DP. CPU:lla on tuki opinnäytetyössä tarvittaville PROFIBUS DP- ja PROFINET I/O-liikennöinneille. Logiikan muut komponentit ovat: virtalähde, digitaalinen lähtö- ja tulokortti ja analogia tulo-/lähtö yhdistelmäkortti. Logiikan komponentit ovat taulukossa 2.

Taulukko 2 Logiikan komponentit

Yksikkö	Tunnus
PS 307	307-1BA01-0AA0
CPU 315 -2 PN/DP	315-2EH14-0AB0
DI 16xDC24V	321-ABH02-0AA0
DO 16xDC24V	322-1BH01-0AA0
AI 4/AO 2 X 8/8 bit	334-0CE01-0AA0

3.4 HMI-PC

Kelaimen käyttöliittymä päätettiin toteuttaa kosketusnäytöllä varustetulla PC:llä. PC:ksi valittiin HP:n All-In-One -tietokone, jossa komponentit on integroitu näytön rakenteisiin ja sitä voidaan ohjata kosketuksella.

3.5 Kireysanturit ja vahvistin

Nauhan kireydenmittaus toteutettiin FMS:n valmistamilla RMGZ 921-pyörivillä kireysmittausantureilla, jotka on liitetty FMS:n EMGZ 521-kireysmittausvahvistimeen. Antureiden mittausalue on 0-100 N ja niiden mittaus-tarkkuus +/- 0.5 %.

Kireysvahvistimessa on kaksi mittauskanavaa ja neljä lähtökanavaa. Vahvistimesta oli mahdollista saada virtaviesti vain yhdestä kanavasta. Tämän vuoksi molemmat mittautiedot tuotiin logiikalle jänniteviestinä. Vahvistimen lähdön erotte-lutarkkuus on 13-bittiä ja mittaussykli 4ms.

4 KÄYTETYT OHJELMISTOT

Laitteiden konfigurointi toteutettiin pääosin PC :llä. Taajuumuuttajien konfigurointiin tarvittiin kaksi ohjelmaa. NXP taajuusmuuttajien konfiguroinnissa käytettiin NCDrive ohjelmaa ja uudemman sukupolven Vacon 20 laitteen konfigurointi toteutettiin Vacon Live ohjelmalla. Logiikka sovellus ohjelmoitiin TIA-portal ympäristössä. Käyttöliittymän ohjelmointi ja kommunikointi logiikan kanssa toteutettiin InTouch ohjelmistolla.

4.1 TIA-Portal

Logiikan ohjelmointi suoritettiin TIA-Portal-ohjelmistolla. TIA-Portal on Siemensin uusi ohjelmointiympäristö, jonka tarkoituksena on tuoda käyttöliittymän, ohjelman ja taajuusmuuttajien ohjelmointi lähemmäksi toisiaan. Tässä projektissa ei hyödytty näistä ominaisuuksista, koska käyttöliittymä toteutettiin InTouchin ohjelmistolla ja taajuusmuuttajat eivät olleet Siemensin omia. TIA-Portal tarjosi kuitenkin helppokäyttöisemmän SCL- ja edistyneemmän GRAPH-editorin S7-manageriin verrattuna. Käytetyllä versiolla oli käytössä FBD, -STL-, SCL- ja GRAPH-ohjelmointikielet. /1/

4.2 InTouch

InTouch-ohjelmisto sisältää useita ohjelmia eri toimintoja varten. Projektissa käytettiin SMC:tä yhteyden hallintaan. Käyttöliittymä ohjelmoitiin Window Maker-ohjelmalla ja käyttöliittymää ajettiin Window Viewer-ohjelmalla.

4.2.1 SMC Orchestra system management console

Orchestra system management console-ohjelman avulla voidaan luoda ja hallita yhteyttä käyttöliittymän ja logiikan välillä. Logiikalla ja käyttöliittymällä on oma liikennöintiprotokollansa ja Orchestra system management consolen avulla voidaan toteuttaa käyttöliittymän ja logiikan välisen liikennöinnin mahdollistava rajapinnan muunnos.

4.2.2 InTouch Window Maker

Window Maker-ohjelmalla ohjelmoitiin käyttöliittymä logiikan ohjausta varten. Ohjelmalla luodaan objekteja ja linkitetään ne tiettyihin logiikan datablokkeihin tai niiden osiin.

4.2.3 Window Viewer

Window Viewer-ohjelma on tarkoitettu käyttöliittymän ajamiseen. Window Viewerillä voidaan käyttää Window Makerilla ohjelmoitua käyttöliittymää.

4.3 NCDrive ja Live

Taajuusmuuttajien parametointia ja monitorointia varten on oma ohjelmansa. NXP-laitteilla kyseinen ohjelma on NCDrive. NCDrivella voidaan tehdä taajuusmuuttajan parametroidit ja suorittaa laitteen monitorointia. Vacon 20 - taajuusmuuttaja edustaa uudempaa sukupolvea ja sille vastaava sovellus on Vacon Live. Ohjelman toiminnot ovat samat kuin NCDrivellä, mutta käyttöliittymä on erilainen.

5 TAAJUUSMUUTTAJIEN PARAMETROINTI

Taajuusmuuttajien parametrointi kuului projektin ensimmäiseen osaan. Työn luonteesta johtuen kaikkea parametrointia ei kuitenkaan voitu tehdä etukäteen. Projektin edetessä tuli muutoksia ja lisäyksiä taajuusmuuttajan ohjausparametreihin.

Pääosin parametrointi suoritettiin kahdessa osassa. Ensimmäiseen osaan kuului taajuusmuuttajan perusparametroidit ja ID-ajon suorittaminen. Toiseen osaan kuului logiikan ja taajuusmuuttajan väliseen liikennöintiin liittyvät parametrit.

5.1 PPO

Logiikan ja taajuusmuuttajan välistä liikennöintiä varten pitää määritellä PPO-tyyppi (Parameter/Process Data Object). PPO:lla määritellään lähetettävän ja vastaanotettavan viestikehyksen koko. Kuviossa 3 on Vaconin PROFIBUS-optiokortin manuaalista kuva käytettävistä PPO -tyypeistä.

The PPOs in Vacon NX:

Parameter Field			Process Data Field										
ID	IND	VALUE	CW	REF	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	
			SW	ACT	PD1	PD2	PD3	PD4	PD5	PD6	PD7	PD8	
PPO 1													
PPO 2													
PPO 3													
PPO 4													
PPO 5													

Kuvio 3 PROFIBUS-väylän PPO-tyypit

Logiikan ja taajuusmuuttajan väliseen liikennöintiin valittiin PPO 5. Kyseisellä PPO-tyypillä voidaan lähettää ja vastaanottaa kahdeksan eri prosessidataa taajuusmuuttajan ja logiikan välillä. Prosessidatan sisältö määritellään linkittämällä taajuusmuuttajan parametrin ID tiettyyn prosessidataan. PPO 5:llä on myös käytössä diagnostiikka kanava, jonka avulla voidaan kirjoittaa tai lukea yksittäisiä parametreja.

Liikennöintiin riitti kahdeksan prosessidataa, joten diagnostiselle kanavalle ei ollut tarvetta. Prosessidatan liikennöintiin vaikuttaa myös optiokortin toimintatila. Toimintatilat ovat: PROFIDRIVE, Bypass ja Echo. PROFIDRIVE on useiden valmistajien yhdessä määrittelemä liikennöintistandardi. Bypass-tilassa liikennöintikortti ei ota kantaa liikennöintidatan sisältöön, vaan se menee muuttamattomana ohjelmalle. Echo-tilassa liikennöintikortti lukee sille lähetetyt prosessidatat ja lähettää ne takaisin logiikalle. Echo-tilassa voidaan testata logiikan ja taajuusmuuttajan välisen liikennöinnin toimivuutta. Opinnäytetyössä liikennöintikorttien tilaksi asetettiin Bypass. /2/

5.2 Prosessidatan määrittely

PPO-tyypillä valittiin taajuusmuuttajan ja logiikan välisen viestin rakenne. PPO 5 sisälsi kahdeksan input- ja output-prosessidataa. Yhdellä prosessidatalla voidaan siirtää logiikan ja taajuusmuuttajan välillä wordin mittainen viesti. Taajuusmuuttajaan parametroidaan mihin parametriin mikäkin prosessidata linkitetään. Kuviossa 4 on esitetty taajuusmuuttajan neljä prosessidatan määrittelyt.

Index	Variable Text	Value	Unit	Min	Max	ID
P 2.13.1	FB Data Out1 Sel	4		n/a	n/a	852
P 2.13.2	FB Data Out2 Sel	1172		n/a	n/a	853
P 2.13.3	FB Data Out3 Sel	1173		n/a	n/a	854
P 2.13.4	FB Data Out4 Sel	1174		n/a	n/a	855
P 2.13.5	FB Data Out5 Sel	1113		n/a	n/a	856
P 2.13.6	FB Data Out6 Sel	6		n/a	n/a	857
P 2.13.7	FB Data Out7 Sel	7		n/a	n/a	858
P 2.13.8	FB Data Out8 Sel	37		n/a	n/a	859
P 2.13.9	FB Data IN 1 Sel	1140		n/a	n/a	876
P 2.13.10	FB Data IN 2 Sel	609		n/a	n/a	877
P 2.13.11	FB Data IN 3 Sel	1278		n/a	n/a	878
P 2.13.12	FB Data IN 4 Sel	1253		n/a	n/a	879
P 2.13.13	FB Data IN 5 Sel	1440		n/a	n/a	880
P 2.13.14	FB Data IN 6 Sel	456		n/a	n/a	881
P 2.13.15	FB Data IN 7 Sel	446		n/a	n/a	882
P 2.13.16	FB Data IN 8 Sel	0		n/a	n/a	883

Kuvio 4 Taajuusmuuttajan neljä prosessidatan määrittelyt

Taajuusmuuttajien prosessidatat olivat pääosin identtisiä, pois lukien muutamaa poikkeusta. Taajuusmuuttajat yksi ja kaksi mittasivat lämpötiloja ja lähettivät mitaustiedot väylän kautta logiikalle. Taajuusmuuttajien kolme ja neljä optiokortin relelähdöt ohjasivat jarruja. Yksi prosessidata määritti milloin CW-bitti 12 ohjasi

relettä ja yksi prosessidata määritti milloin taajuusmuuttajan jarrulogiikka ohjasi relettä.

5.3 CW ja SW

Taajuusmuuttajan hallinta toteutetaan CW:llä (control word) ja taajuusmuuttajan tilatiedot saadaan SW:stä (status word). Control wordin rakenne riippuu valitusta PROFIBUS-optiokortin toimintatilasta ja taajuusmuuttajan sovelluksessa käytetävästä tilakoneesta. Opinnäytetyössä optiokortilla käytettiin Bypass-tilaa ja taajuusmuuttajan sovelluksen tilakoneena oli PROFIDRIVE. Tällä yhdistelmällä käytössä oleva CW ei kuitenkaan vastaa Profibus standardia vaan se on määritelty taajuusmuuttajan sovelluksessa. Profidrive 2.0 mukaisen tilakoneen ja CW saa käyttöön valitsemalla optiokortin toimintatilaksi profidrive ja ohjelman tilakoneeksi standardidaan. Kuviossa 5 on taajuusmuuttajan ohjelmamanuaalissa oleva CW:n rakenne Bypass-PROFIDRIVE-yhdistelmälle.

FB Control Word		
	Signal	Comment
b0	ON	0>1 will reset the Switch On Inhibit state and bring the drive to Ready Run. Should be reset after fault, Coast Stop (b1) and Emergency Stop (b2) .
b1	Coasting Stop	0=Coast stop Active 1=Coast stop NOT active
b2	Quick Stop	0=Quick stop Active 1=Quick stop NOT active
b3	Start	Normal start command 0=Stop the drive 1=Start the drive
b4	Ramp Output to Zero	0=Force speed ramp output to zero 1=Release speed ramp output
b5	Ramp Hold	0=Hold speed ramp output 1=Release speed ramp
b6	Ramp Input to Zero	0=Force speed ramp Input to zero 1=Release speed ramp Input
b7	Fault Reset	0=No Action 1=Reset active faults
b8	Inching 1	Run the drive with defined constant speed 0=No Action 1=Run with constant speed
b9	Inching 2	Run the drive with defined constant speed 0=No Action 1=Run with constant speed
b10	Fieldbus Control Enable	Activate Fieldbus control when P3.1 =3/Fieldbus 0=Fieldbus Control NOT active 1=Activate Fieldbus Control
b11	Watch Dog	0>1>0>1...1 sec square wave clock. This is used to check data communication between Profibus master and the drive. Used to generate FB Communication -Fault.
b12		
b13		
b14		
b15		

Kuvio 5 Bypass-PROFIDRIVE CW

CW-bitit 12–15 ovat vapaasti valittavissa omiin ohjauksiin ja niiden toiminto voidaan määritellä NCDrivella. Kelojen jarruja ohjattiin taajuusmuuttajan optiokortin releellä. Projektin alussa oli tarkoitus, että jarrujen ohjaus toteutettaisiin taajuusmuuttajan jarruohjaustoiminnolla. Tällä ei kuitenkaan saavutettu haluttua nauhan kireyttä pysähdysten aikana, joten jarrujen ohjaus siirrettiin logiikalla. Tämä tehtiin määrittelemällä CW-bitti 12 ohjaamaan optiokortin relettä. CW:n vapaat bitit linkitettiin toimintoihin kuvion 6 parametreilla.

P 2.5.1.17	Limit Control UN	DigOUT:0.1		n/a	n/a	454
P 2.5.1.18	FB Dig Input 1	DigOUT:0.1		n/a	n/a	455
P 2.5.1.19	FB Dig 1 Par ID	0		n/a	n/a	891
P 2.5.1.20	FB Dig Input 2	DigOUT:0.1		n/a	n/a	456
P 2.5.1.21	FB Dig 2 Param.	0	ID	n/a	n/a	892
P 2.5.1.22	FB Dig Input 3	DigOUT:0.1		n/a	n/a	457
P 2.5.1.23	FB Dig 3 Param.	0	ID	n/a	n/a	893
P 2.5.1.24	FB Dig Input 4	DigOUT:0.1		n/a	n/a	169
P 2.5.1.25	FB Dig 4 Param.	0	ID	n/a	n/a	894
P 2.5.1.26	FB Dig Input 5	DigOUT:0.1		n/a	n/a	170
P 2.5.1.27	FB Dig 5 Param.	0	ID	n/a	n/a	895
P 2.5.1.28	Safe Disable Act	DigOUT:0.1		n/a	n/a	756
P 2.5.1.29	MCC Close Cont.	DigOUT:0.1		n/a	n/a	1218
P 2.5.1.30	MCC Close Pulse	DigOUT:0.1		n/a	n/a	1219
P 2.5.1.31	Motor Fan Cont.	DigOUT:0.1		n/a	n/a	1805

Kuvio 6 CW-bit12 määrittely

Kuviossa 6 näkyvät FB Dig inputit ja Dig Paramet ID:t linkittyvät CW:n vapaisiin bitteihin. FB Dig 1 on oletuksena varattu watch dogille, joten ensimmäinen vapaa ohjelmoitava bitti on FB Dig Input 2. FB Dig Inputilla voidaan määrittää bitti ohjaamaan haluttua optiokortin lähtöä tai FB Dig-parametrilla ohjaamaan määritettyä parametria päälle tai pois. Status Wordin rakenteeseen vaikuttaa myös liikennöintikortin toimintatila ja taajuusmuuttajan sovelluksen käyttämä tilakone. Kuviossa 7 taajuusmuuttajan sovelluksen manuaalin selvitys SW-rakenteesta Bypass-PROFIDRIVE-tilassa.

	FB Status Word	
	Signal	Comment
b0	Ready to switch On	0=The drive NOT ready to switch ON 1=The drive is ready to switch ON
b1	Ready to Operate	0=The drive is NOT ready to run 1=The drive is ready to run
b2	Running	0=The drive is NOT running 1=The drive is running and ready to release the reference
b3	Fault Active	0=No fault active 1=Fault IS active
b4	Cost Stop NOT active	0=Coast stop active 1=Coast stop NOT active
b5	EM Stop not Active	0=Emergency stop active 1=Emergency stop NOT active
b6	Switch On Inhibit	0=No Inhibit 1=The drive is out of fault and coast / emergency stop state.
b7	Warning	0=NO alarm 1=Alarm IS active
b8	Speed At Ref	0=Speed actual is NOT equal to speed reference 1=Speed actual is equal to speed reference
b9	FB Control Active	0=Fieldbus Control NOT active 1=Fieldbus Control active
b10	Above Limit	Indicate if speed actual is below the limit P2.4.16 0=Speed actual is below the speed limit 1=Speed actual is above the speed limit
b11	SW ID.Bit selection B11	P2.13.22 SW B11 ID.Bit
b12	SW ID.Bit selection B11	P2.13.23 SW B12 ID.Bit
b13	SW ID.Bit selection B11	P2.13.24 SW B13 ID.Bit
b14	SW ID.Bit selection B11	P2.13.25 SW B14 ID.Bit
b15	Watch Dog Feedback	

Kuvio 7 SW Bypass-PROFIDRIVE-tilassa

Myös SW:ssa on mahdollista parametroida vapaat bitit 11–14 indikoimaan haluttuja tietoja. Käytettäessä PROFIDRIVE-tilakoneetta, SW-bitit yksi ja kolme antavat tilatietoja PROFIDRIVE-tilakoneelta. Tilakone antoi ready to operate-tiedon vasta, kun taajuusmuuttaja oli saanut käynnistyskäskyn. Warning-indikointia tilakoneelta ei saatu olleenkaan. SW- bitti 11 linkitettiin taajuusmuuttajan ohjelman tilakoneeseen indikoimaan ready to operate-tilaa. SW-bitti 12 linkitettiin indikoimaan taajuusmuuttajan ohjelman tilakoneelta saatavaa varoitusta. Kuviossa 8 on SW-bittien määrittely.

P 2.13.19	State Machine	2 / ProfiDrive		n/a	n/a	896
P 2.13.20	FB Ref Filter TC	0	ms	n/a	n/a	863
P 2.13.21	FB Monitoring	1 / Yes		n/a	n/a	1629
P 2.13.22	SW B11 ID.Bit	43,01		n/a	n/a	1625
P 2.13.23	SW B12 ID.Bit	43,07		n/a	n/a	1626
P 2.13.24	SW B13 ID.Bit	0,00		n/a	n/a	1627
P 2.13.25	SW B14 ID.Bit	0,00		n/a	n/a	1628

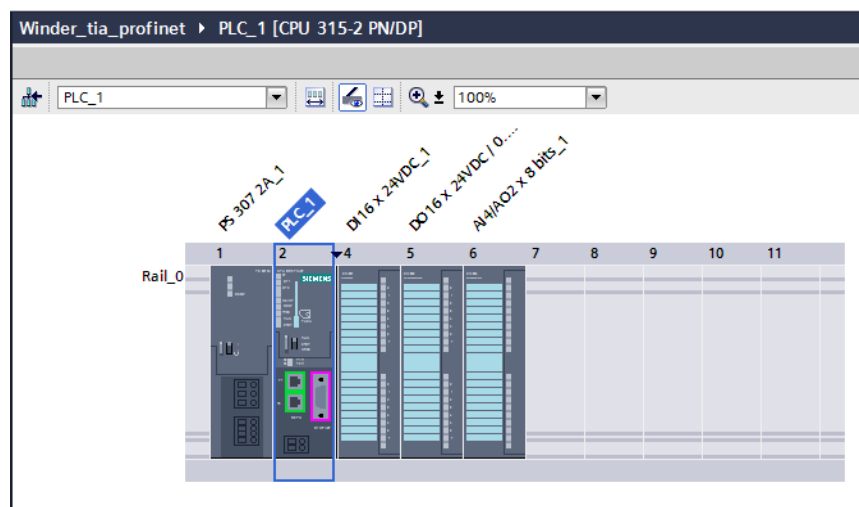
Kuvio 8 SW-bittien 11 ja 12 määrittely

6 LOGIIKAN OHJELMOINTI

Logiikan ohjelmointi suoritettiin Siemensin S7 TIA-Portal-ohjelmistolla. Käytettyjä ohjelmointikieliä olivat SCL, FBD ja GRAPH. SCL:llä ohjelmoitiin suurin osa matemaattisista ja tekstiä sisältävistä lohkoista. GRAPH:a käytettiin sekvenssien tekemiseen ja FBD:llä ohjelmoitiin loput. Ohjelmointi aloitettiin hardware-määrittelyillä. Tämän jälkeen ohjelmoitiin taajuusmuuttajan tyyppilohko, jotta voitiin tarkastaa taajuusmuuttajien ja logiikan välisen yhteyden toimivuus. Taajuusmuuttajien ja logiikan välisen yhteyden oltua kunnossa, ohjelmoitiin laskentalohkot ja sekvenssit.

6.1 Hardware-asetukset

TIA-Portalilla aloitettiin uusi projekti ja siihen lisättiin taulukon 2 mukaiset komponentit. Kuviossa 9 on hardware-asetuksissa määritelty logiikan kokoonpano.



Kuvio 9 Logiikan komponentit

Tämän jälkeen määriteltiin CPU-yksikön liikennöintiasetukset. Kuviossa 10 näkyy logiikalle määritelty aliverkko, IP-osoite ja aliverkon peite.

PROFINET interface_1 [PN-IO]

General IO tags Texts

General

Ethernet addr...

Time synchro...

Operating mode

Advanced opt...

Interface o...

Media redun...

Real time ...

Port [X2 P1 R]

Port [X2 P2 R]

Diagnostics a...

Ethernet addresses

Interface networked with

Subnet: PN/IE_1

Add new subnet

IP protocol

☒ Set IP address in the project

IP address: 192 . 168 . 0 . 1

Subnet mask: 255 . 255 . 255 . 0

☐ Use router

Router address: 0 . 0 . 0 . 0

☐ Set IP address using a different method

PROFINET

☐ Set PROFINET device name using a different method.

☒ Generate PROFINET device name automatically

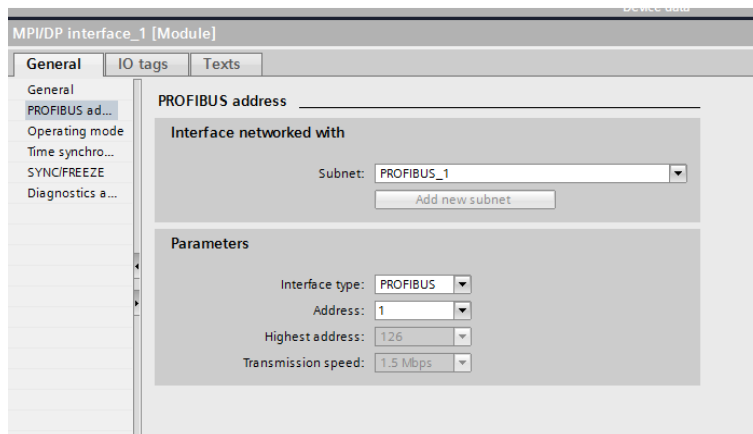
PROFINET device name: plc_1

Converted name: plcx1d0ed

Device number: 0

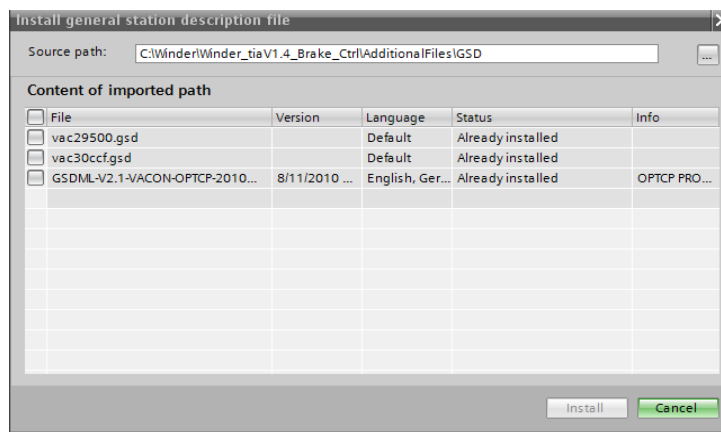
Kuvio 10 Ethernet -asetukset

Kuviossa 11 on logiikalle määritellyt PROFIBUS-asetukset. Aliverkoksi on valittu PROFIBUS ja logiikan osoitteeksi on määritetty 1. Operating mode-valikosta logiikan toimintatilaksi on valittu master.



Kuvio 11 PROFIBUS-liikennöintiasetukset

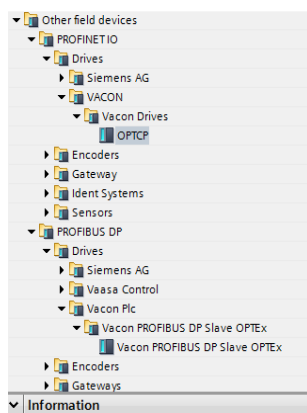
Seuraavaksi hardware-asetuksiin lisättiin taajuusmuuttajat. Jotta logiikka ja taajuusmuuttajat voisivat liikennöidä keskenään, pitää logiikkaan asentaa toimilaittevalmistajan tekemä GSD- ja GSDML-tiedosto. PROFIBUS-liikennöinnissä GSD-tiedosto sisältää laitekohtaiset tiedot, joilla mahdollistetaan toimilaitteen ohjaaminen I/O-laitteen tavoin. PROFINET I/O:ta käytettäessä vastaava tiedosto on GSDML. Kuviossa 12 on työssä käytetyt GSD- ja GSDML-tiedostot.



Kuvio 12 GSD- ja GSDML-tiedostot

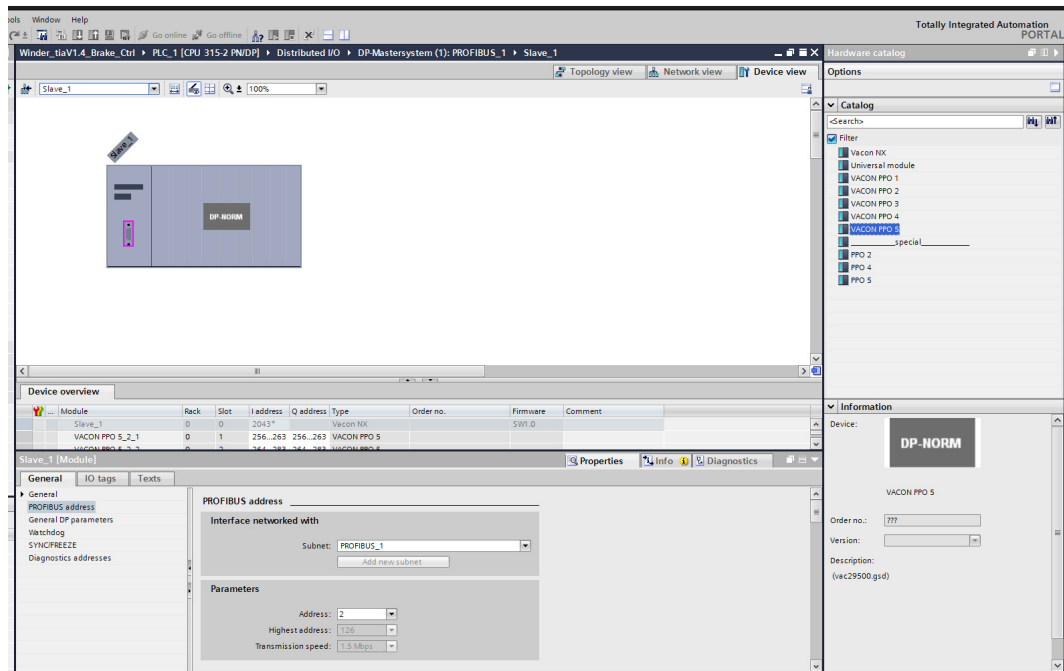
Tiedostojen asennuksen jälkeen toimilaitelistalta tuodaan projektiin PROFINET I/O:n puolelta neljä Vaconin OPTCP-korttia ja PROFIBUS DP:n puolelta neljä NXP:n PROFIBUS DP slave-laitetta ja yksi Vacon 20 PROFIBUS DP slave. NXP-taajuusmuuttajia on tarkoitus ohjata vaihtoehtoisesti PROFIBUS:in tai PROFI-

NET I/O:n välityksellä. Tämän vuoksi hardware-asetuksissa on kaksi laitetta yhtä NXP-taajuusmuuttajaa kohden. Toimilaitelista on kuviossa 13.



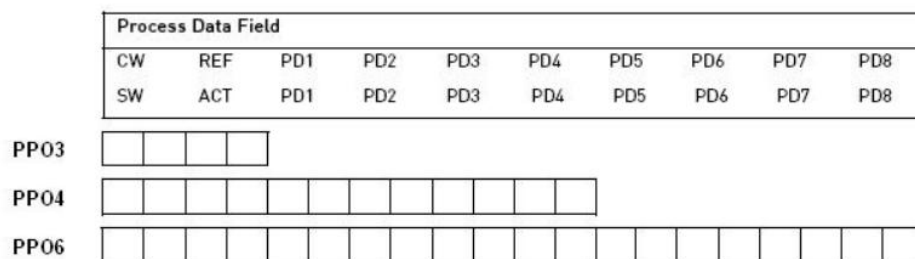
Kuvio 13 Toimilaitelista

Tämän jälkeen määriteltiin liikennöintikorttien asetukset. Ensin määriteltiin taajuusmuuttajien PROFIBUS-asetukset. Samoin, kuin taajuusmuuttajan päässä, myös logiikan päässä pitää määritellä PPO-tyyppi. Logiikalle valitaan sama PPO 5-tyyppi mitä käytetään taajuusmuuttajalla. Tämän lisäksi laitteelle pitää määritellä subnet ja slave-numero. Subnetillä määritellään mitä PROFIBUS-porttia käytetään liikennöidessä laitteen kanssa. Slave-numerolla identifioidaan viesti tietyllä väylässä olevalle laitteelle. Sama slave-numero pitää määritellä myös taajuusmuuttajan PROFIBUS-optiokortille. Tuulettimia ohjaavaa Vacon 20:stä ohjattiin vain PROFIBUS-väylän kautta. Laitteelle valittiin PPO3 sillä, sille annettiin vain käynnistyskäsky ja taajuusohje. Kuviossa 14 on NXP:n PROFIBUS-asetukset logiikan puolelta.



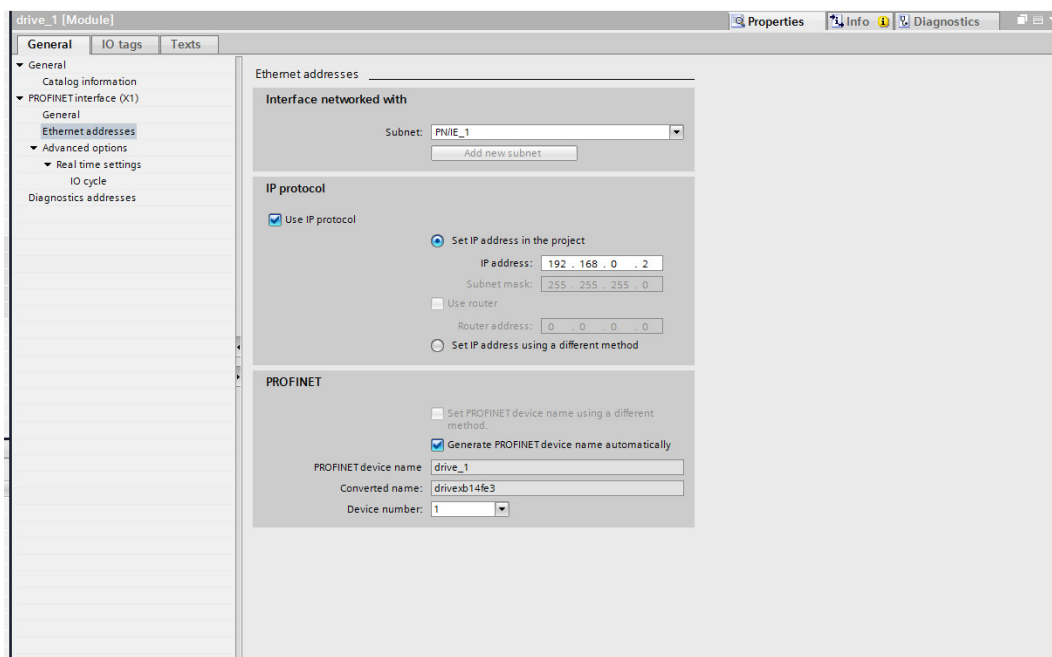
Kuvio 14 PROFIBUS slave-asetukset

PROFINET I/O:ta käytettäessä laite näkyy ohjelmoinnin kannalta logiikan päässä samalla tavalla, kuin PROFIBUS:a käytettäessä. Myös PROFINET I/O:ssa laitteelle pitää määritellä PPO-tyyppi, jotta logiikka pystyy kommunikoimaan taajuusmuuttajan kanssa. PROFINET I/O:n PPO-tyypit ovat rakenteeltaan erilaisia kuin PROFIBUS:in. Kuviossa 15 on PROFINET I/O:n PPO-tyyppien rakenteet. PPO6 sisältää yhtä monta prosessidataa kuin PROFIBUS:in PPO5. Erona on diagnostiikkakanavan puuttuminen, joka on toteutettu PROFINET:ssä eri tavalla kuin PROFIBUS:ssa. PROFINET I/O:ta käytettäessä pitää myös valita mitä toimintatilaa PPO:n kanssa käytetään. Työssä käytettiin PROFINET I/O:n puolella PPO6:sta Bypass-tilassa.



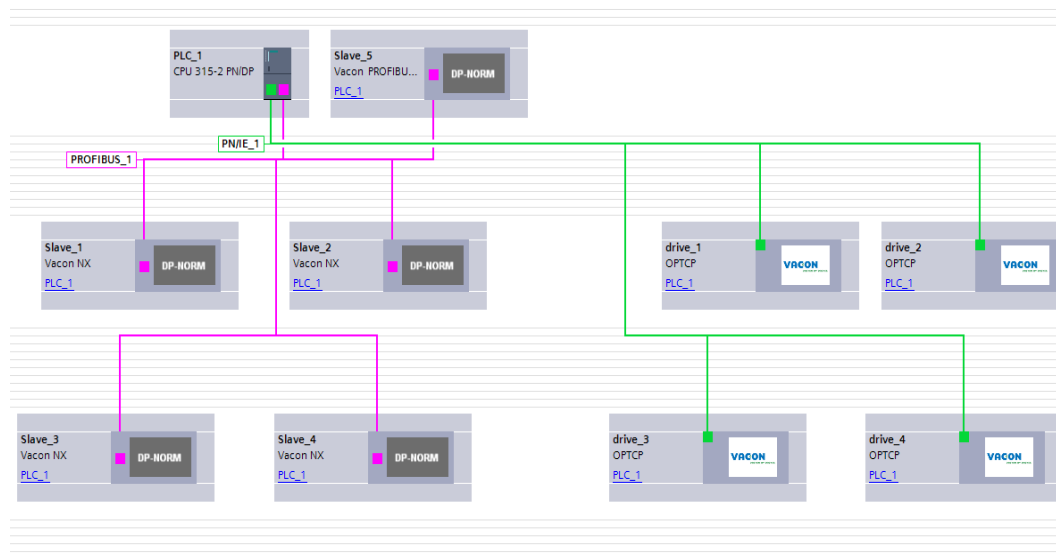
Kuvio 15 PROFINET I/O PPO-tyypit

PPO:n lisäksi PROFINET I/O-liikennöintiä varten laitteille piti määrittellä aliverkko, IP-osoite ja PROFINET-laitenimi. Kuviossa 16 on määritelty taajuusmuuttajan aliverkko ja IP-osoite. IP-osoite ja aliverkonpeite pitää määrittellä myös taajuusmuuttajan PROFINET-kortille.



Kuvio 16 Taajuusmuuttajan PROFINET I/O-asetukset

Laitteiden IP-osoitteen ja aliverkonpeitteen määrittelyn jälkeen siirryttiin online-tilaan, ja määriteltiin taajuusmuuttajille PROFINET-liikennöintiä varten tarvittavat laitenimet. Kaikille neljälle NXP-taajuusmuuttajalle tehtiin samat asetukset. Slave-numero, IP-osoitteen viimeinen osa ja laitenimi olivat kuitenkin laitekohtaisia. Kuviossa 17 on logiikan hardware-asetuksen verkkorakenne. Slave-numerot 1-4 ja drive 1-4 ovat NXP-taajuusmuuttajia ja slave-numero 5 on Vacon 20. Sama slave- ja drive-numero viittaa samaan laitteeseen.



Kuvio 17 Logiikan verkkoasetukset

6.2 Taajuusmuuttajan tyyppilohko

Taajuusmuuttajan ohjauksen pystyy toteuttamaan eri tavoilla. CW voidaan kirjoittaa PROFINET-ohjekirjan mukaisesti hex-lukuna tai sitten kontrolloida yksittäisiä bittejä muodostaen niistä CW. Työssä piti hallita viittä taajuusmuuttajaa. Tämän todettiin onnistuvan parhaiten tekemällä tyyppilohko, jossa muodostetaan CW yksittäisistä biteistä. Ohjelmalohkon pohjana käytettiin koulussa tehtyä taajuusmuuttajan ohjauslohkoa. Ohjelmalohkoon tehtiin muutamia muutoksia, jotta se vastaisi työn vaatimuksia. Alkuperäisessä lohkoissa oli jo kaikki tarvittava taajuusmuuttajan perushallintaan. Siihen lisättiin vielä momenttirajanohjaus, jarrunohjaus ja varoituksenindikointi. SW-bittien tulkintaan tehtiin muutos Ready-indikoinnin kohdalla. Indikointi luettiin ohjelman tilakoneesta ja linkitettiin SW-bittiin 12. Kuviossa 18 on valmis taajuusmuuttajan tyyppilohko.



Kuvio 18 Taajuusmuuttaja tyyppilohko

6.3 Laskentalohkojen ohjelmointi

Ohjelmassa on paljon laskentaa, joten helpoin tapa niiden toteuttamiseksi oli tehdä valmiita laskentalohkoja. Lohkoille syötetään tarvittavat tiedot ja ne antavat vastauksen. Tällä tavoin pääohjelman rakenteesta saadaan selkeämpi ja helpommin hallittava.

6.3.1 Voiman mittaus

Voiman mittaus tuli logiikalle analogiatietona ja se piti skaalata Newtonieiksi. Mittavahvistimelta tulevan tiedon piti olla 0-100 N / 0-10 V, mutta analogiatieto oli skaalaantunut eri tavoin. Laskentalohkoon piti tehdä oma skaala, jotta mittaus saatiin oikeaksi. Skaalauksessa käytettiin apuna punnuksia, joiden paino oli tunnettu. Kuviossa 19 on voiman mittauksen skaalauslohko.

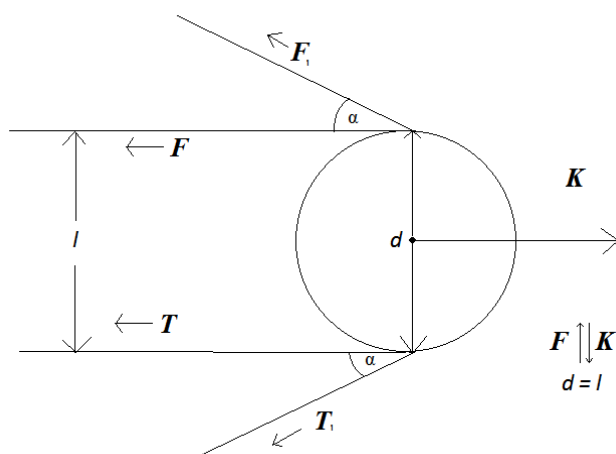
```

0001 //channel 1
0002
0003 #help1:=WORD_TO_DINT(#Fmeas_Ch1);
0004 #help2:=DINT_TO_REAL(#help1);           //converting measurement word->real
0005 #Ch1_F:=(#help2/27648.0)*400.0;         //scaling measurement 0-100N
0006
0007 //channel 2
0008 #help3:=WORD_TO_DINT(#Fmeas_ch2);
0009 #help4:=DINT_TO_REAL(#help3);
0010 #Ch2_F:=(#help4/27648.0)*400.0;

```

Kuvio 19 Voiman mittauksen skaalaus

Nauha kiersi pyörivän voiman mittausanturin ympäri, joten mitattu voima ei vastannut suoraan nauhan kireyttä. Nauha pyrkii liikuttamaan anturia ja siihen voitiin soveltaa liikkuvan väkipyörän yhtälöitä. Kuviossa 20 nauhan kireyttä kuvaavat voimavektorit T_1 ja F_1 . Anturin lähettämää voimatietoa kuvaa K .



Kuvio 20 Väkipyörän voimat

Mikäli nauhan kireyttä kuvaavat voimavektorit eivät ole vastakkaissuuntaisia, kuten kuvion 20 F_1 ja T_1 , pitää lasketut T ja F jakaa $\cos\alpha$:lla, jotta saadaan laskettua nauhan suuntainen voimavektori. /3/

Kelaimessa nauhojen välinen etäisyys toisistaan oli lähes sama kuin voiman mittausanturin halkaisija. Voitiin siis olettaa nauhaan kohdistuvien voimavektoreiden olevan vastakkaissuuntaisia anturin voimavektorin suhteen. Lopullinen mitattu nauhan kireys saatiin jakamalla mitattu voima kahdella. Voimanmittauksessa oli paljon heittelyä. Tämän vuoksi mittauksesta otettiin vielä kaksi eri liukuvaa keskiarvoa. Keskiarvot laskettiin viidellä ja 20 näytteellä.

6.3.2 Nopeuden mittaus

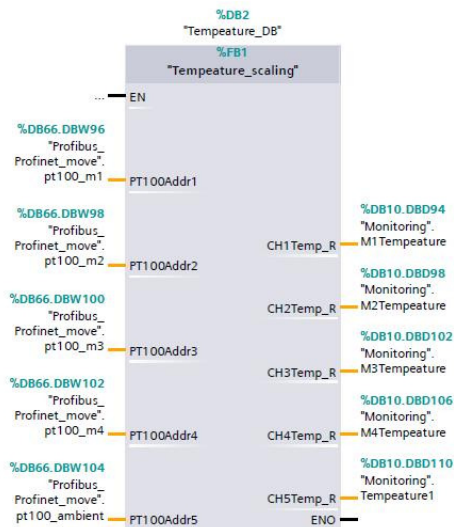
Nauhan nopeus laskettiin moottoreiden yksi ja kaksi pyörimisnopeuksien avulla. Moottoreiden nopeudesta laskettiin keskiarvo ja se kerrottiin telojen ympärysmittalla. Kuviossa 21 on nauhan nopeuden laskenta.

```
0001 //calculating webspeed from drives 1 and 2 average
0002
0003 #avg1:=(#Actspeak1+#Actspeak2)/2;
0004 #avg2:=#avg1;
0005 #Webspeed_rpm:=(#avg1+#avg2)/2;
0006 #Webspeed:= "Parameters".Emptyreeldiameter*3.14*(#avg1+#avg2)/2;
0007
0008
```

Kuvio 21 Nauhan nopeuden laskentalohkon ohjelmakoodi

6.3.3 Lämpötilojen mittaus

Lämpötilojen mittaustieto tuli taajuusmuuttajilta väylän kautta logiikalle ja se skaalattiin vastaamaan celsius-asteita. Lohko oli varsin yksinkertainen. Sisään tuleva tieto muutettiin real-luvuksi ja jaettiin kymmenellä vastaamaan oikeaa lämpötilaa. Kuviossa 22 on toteutettu lämpötilan mittauslohko.



Kuvio 22 Lämpötilan mittauslohko

6.3.4 Halkaisijan laskenta

Auki- ja kiinnikelaimen halkaisija voitiin laskea niiden pyörimisnopeuden suhteesta nauhan nopeuteen kaavalla 2.

$$d = \frac{v}{n \cdot \pi} \quad (2)$$

Nauhan nopeuden yksikkö on metriä minuutissa ja pyörimisnopeuden yksikkö on kierrosta minuutissa. Aikayksiköitä ei tarvinnut muuttaa sekunneiksi, vaan halkaisija voidaan laskea suoraan syötetyillä arvoilla. Kuviossa 23 on halkaisijan laskenta.

```
0001 #Diameter:=#WebSpeed/ (#RPM*3.14); //calculating diameter from web speed and mo-
tors speed
```

Kuvio 23 Halkaisijan laskentalohkon ohjelmakoodi

Halkaisijan laskennasta saatu tulos käsiteltiin vielä rampilla. Rampin avulla saatiin rajoitettua halkaisijan laskentatuloksen maksimi muutosnopeus vastaamaan teoreettista maksimia. Rampin nousunopeus (slope) määriteltiin kelan maksimi pyörimisnopeudesta ja nauhan paksuudesta. Rampin nousunopeus oli kiinteä arvo, mikä aiheutti rajoitukseen epätarkkuutta. Kelan halkaisijan ollessa iso, sen pyörimisnopeus on pienempi. Myös halkaisijan muutosnopeus on huomattavasti pie-

nempi. Kiinteä rampin nousunopeus mahdollisti halkaisijan laskennan tuloksen muuttumisen samalla nopeudella kelan pyörimisnopeudesta riippumatta. Tämän vuoksi rampin käyttö ei riittänyt estämään nopeudenheittelystä ja mittausvirheistä aiheutuvaa halkaisijan laskennan virhettä.

Laskentaa tarkennettiin ottamalla syötetyistä nopeuksista 20 näytteen liukuvakeskiarvo. 20 ms ohjelmakierrolla keskiarvon laskenta aiheutti 400 ms viiveen halkaisijan laskentaan, mutta sillä saatiin suodatettua mittausvirheet ja nopeuden heilahtelut. Tällä tavoin saavutettiin tarkempi halkaisijan laskenta, jonka tuloksen heilahtelu ajon aikana oli noin +/- 1mm.

6.3.5 Nauhan pituuden laskenta

Nauhan pituus voidaan laskea, kun tiedetään kelan halkaisija, puolan halkaisija ja nauhan paksuus. Tässä lohkossa käytetään puolan halkaisijana kelojen minimihalkaisijaksi asetettua arvoa. Tällä tavoin saadaan laskettua kuinka paljon kelalla on nauhaa jäljellä automaattiseen pysäytyskohtaan. Nauhan pituus per kierros muurruu joka kierroksella saman verran ja näin ollen muodostaa aritmeettisen lukujonon. Aritmeettinen lukujono voidaan laskea summalausekkeella tai siitä johdetun kaavan 3 mukaan. Kaavassa 3, S_n on nauhan kokonaispituus, n on kierrosten määrä, max on suurin laskettu kehän ympärysmitta ja min on pienin laskettu kehän ympärysmitta.

$$S_n = \frac{n*(max+min)}{2} \quad (3)$$

Kierrosten määrä voidaan laskea jakamalla säde nauhan paksuudella. Kuviossa 24 on pituuden laskentakoodi.

```

0001 #Revolutions:=(#Diameter-#Min_Dia)/(2*#Web_Thicknes); //calculating revolutions
    in reel
0002 #MinCircumference:=#Min_Dia*3.14; //calculating shortest circumference
0003 #MaxCircumference:=#Diameter*3.14; //calculating longest circumference
0004 //calculating length with arithmetic sum sn=((a1+an)/2)*n
0005 #length:=((#MinCircumference+#MaxCircumference)/2)*#Revolutions;

```

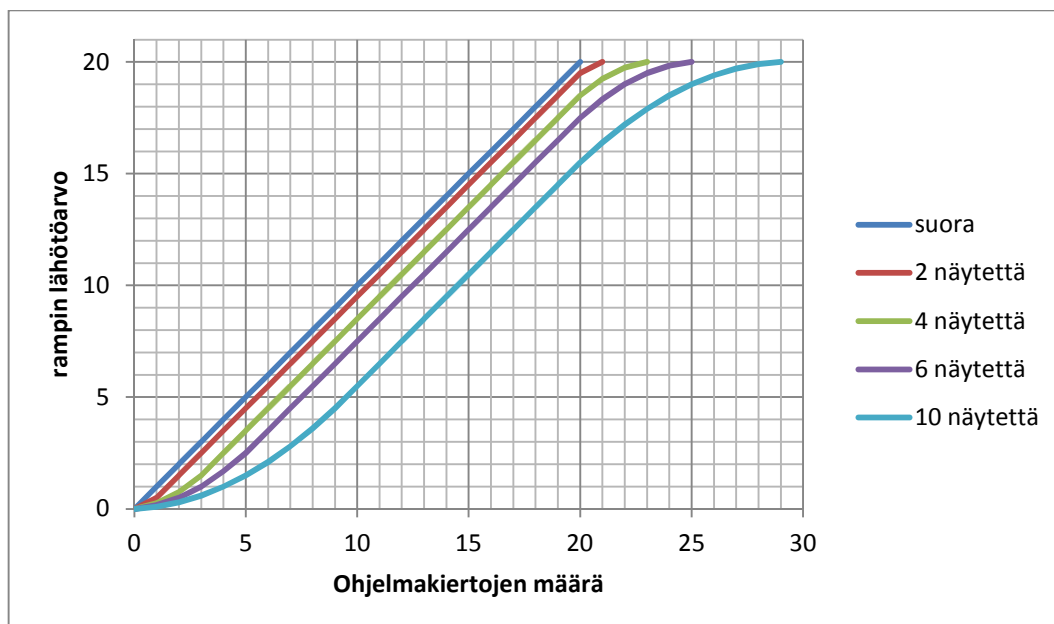
Kuvio 24 Pituuden laskentalohkon ohjelmakoodi

6.3.6 Pituusennakon laskenta

Taajuusmuuttajille annetaan nopeusohje rampin kautta. Tämän vuoksi prosessin alasajoon kuluva aika riippuu linjanopeudesta. Kelaimen automaattinen pysäyttäminen oikeaan kohtaan linjanopeudesta riippumatta, vaatii tiedon kuinka paljon nauhaa linjan läpi pysäytyksen aikana menee. Kun tiedetään linjamootoreiden sen hetkinen pyörimisnopeus ja rampin nousunopeus(slope), voidaan laskea kuinka monta sekuntia kuluu nollanopeuden saavuttamiseen.

Pysäytyksen alun linjanopeuden ja pysäytykseen kuluvan ajan avulla voidaan laskea kuinka monta kierrosta moottorit pyörivät pysäytyksen aikana. Pysäytyksen aikaiset kierrokset voidaan laskea integroimalla pyörimisnopeus ajan suhteen. Linjatelojen ympärysmitan ja pysäytyksen aikana kuluvien kierrosten avulla voidaan laskea, kuinka paljon nauhaa pysäytyksen aikana linjan läpi menee. Nopeusohjeen muutos ei kuitenkaan ole lineaarinen vaan se on pyöristetty alku ja loppupäästä, minkä vuoksi

Rampin pyöristys tehdään laskemalla nopeusohjeesta liukuva keskiarvo. Tämä lisää nopeudenmuutokseen kuluva aikaa. Pyöristyksen aiheuttama lisäaika riippuu ohjelmakiertoajasta ja liukuvan keskiarvon näytteiden määrästä. Lopullinen nopeus saavutetaan vasta, kun keskiarvolohkon kaikki näytteet ovat saaneet saman arvon. Kuviossa 25 olevat kuvaajat näyttävät keskiarvolaskennalla toteutetun rampin pyöristyksen aiheuttaman viiveen loppunopeuden saavuttamiseen. Käyrät on piirretty keskiarvolaskulla saatujen tulosten taulukoiduista arvoista. Käyrästä voidaan huomata, että keskiarvolaskennan aiheuttama viive keskiarvolaskennan aiheuttama viive voitiin laskea seuraavalla tavalla: $Programcycletime \cdot (AVGsamplecount - 1)$.



Kuvio 25 S-rampin viive

Ohjelmassa rampin pyöristys lasketaan 20 näytteellä ja ohjelmakiertosykli on 20 ms, joten laskettuun pysäytykseen kuluvaan aikaan lisätään 380 ms.

6.3.7 Pysäytysten aikaisten kierrosten laskeminen

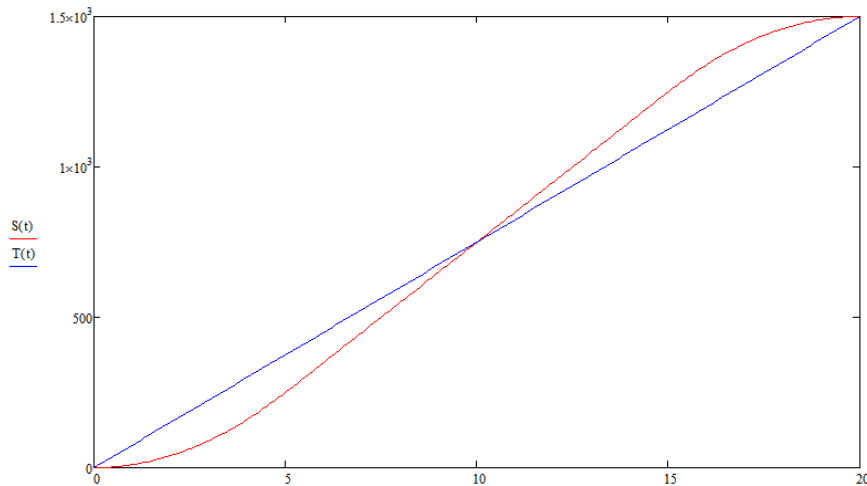
Integraalin laskenta kuluttaisi turhaa CPU:n laskentatehoa ja on kannattavampaa laskea pysäytyksen aikaiset kierrokset kolmion pinta-alana. Pyöristys muuttaa rampin muotoa, mutta se on symmetrinen molemmista päistä. Kuviossa 26 on Mathcadilla toteutettu vertailu symmetrisen s-käyrän ja suoran integraalien välillä. Vertailusta voimme huomata s-käyrän ja suoran määrätyn integraalin erotuksen olevan nolla.

$$s_1 := 5 \quad s_2 := 5 \quad t_t := 20 \quad t := 0, 0.1..t_t \quad n_N := 1500 \quad n_a := 0 \quad k_1 := \frac{n_N - n_a}{2 \cdot s_1 \cdot (t_t - s_2)} \quad k_2 := \frac{n_N - n_a}{2 \cdot s_2 \cdot (t_t - s_1)}$$

$$S(t) := \left[n_a + (k_1 \cdot t^2) \cdot (t < s_1) \right] + \left[\frac{t - s_1}{t_t - s_1} \left[n_N - n_a - (k_1 \cdot s_1^2) + (k_2 \cdot s_2^2) \right] + (k_1 \cdot s_1^2) \right] \cdot (t - s_1 \geq 0) \cdot [t \leq (t_t)] - \left[k_2 \cdot [t - (t_t - s_2)]^2 \right] \cdot (t > t_t - s_2)$$

$$T(t) := \frac{n_N}{t_t} \cdot t$$

$$\int_0^{20} S(t) dt - \int_0^{20} T(t) dt = 0$$



Kuvio 26 S-käyrän ja suoran vertailu

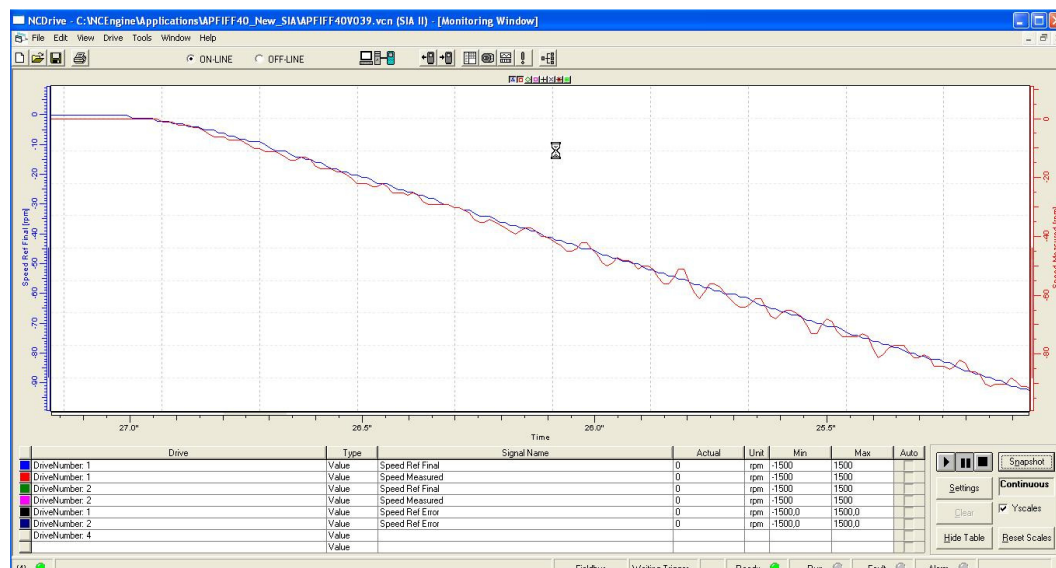
Halkaisijan laskennassa käytetään liukuvaa keskiarvoa, jotta laskentaan ei tulisi turhaa heiluntaa. Tämän vuoksi aukikelaimen laskettu halkaisija on hieman suurempi, kuin todellinen. Tätä kompensoitiin lisäämällä pituusennakoinnin aika-akselin laskentaan 10 ohjelmakierron aika. Lisätty aika määriteltiin koeajon perusteella. Kuviossa 27 on pituusennakon laskennan ohjelmakoodi.

```
0001 #RampTime:=#DrivePV/((1/#Cycle)*#RampSlope)+(10+#AVGCount)*#Cycle; //counting
    how long takes to ramp to zero
0002 #"Speed_M/s":=(#DrivePV/60)*(#ReelDia*3.14); //calculating web-
    speed from rpm
0003 #WebLenght:=(#"Speed_M/s"*#RampTime)/2;
0004 //weblenght prediction is calculated by triangels surfacearea. Surface area of
    triangle is same as integral
0005 //of s-curved ramp when s-ramp is symmetrical. Time axel also needs to include
    additional time average
0006 //counting takes. added 10ms time to compensate diametercalculations averaging.
```

Kuvio 27 Pituusennakkolohkon ohjelmakoodi

Tarkan pysäytyskohdan saavuttaminen oli hankalaa, koska halkaisijan laskenta heittelee. Myös taajuusmuuttajan nopeus ei täysin vastaa nopeusohjetta, jolloin ennakoinnin laskenta ei vastaa toteutunutta. Kuviossa 28 on NCDrivella mitattu

nopeus ja annettu nopeusohje pysäytyksen aikana. Koeajoissa kelain saatiin kuitenkin pysähtymään asetettuun halkaisijaan ± 3 mm heitolla.



Kuvio 28 Taajuusmuuttajan nopeusohje ja toteutunut nopeus

6.3.8 Kelan massan laskenta

Massahitausmomenttia ja kitkan kompensointia varten tarvittiin kelan massa. Kelan massa saatiin laskettua halkaisijan, tyhjän kelan ja täyden kelan painon avulla. Täysi ja tyhjä kela punnittiin, jotta voitiin laskea ominaispaino käytetylle lasikuitunauhalle. Nauhan leveys on vakio, joten lohkon laskuissa ei oteta huomioon kuin kaksi ulottuvuutta. Lohko laskee kelan sen hetkisen halkaisijan ja minimihalkaisijan avulla lasikuitunauhan pinta-alan. Ominaispainon avulla tiedettiin pinta-alan ja massan suhde ja voitiin laskea kelan massa. Kuviossa 29 on massan laskennan ohjelmakoodi.

```

0001 #Area:=(3.14*(#Radius*#Radius))-(3.14*(#Minradius*#Minradius)); //reels area
0002 #Mass:=#Area*#Rho+#EmptyReelMass; //multiplying area with Rho(kg/m2) volume is
    not calculated because z-dimensions is static
0003

```

Kuvio 29 Massan laskentalohkon ohjelmakoodi

6.3.9 Derivointi

Hitausmomentin kompensointia varten pitää laskea kulmanopeuden derivaatta. Derivaatalla voidaan määritellä suureen paikallinen muutosnopeus suhteessa toiseen suureeseen. Tässä tapauksessa haluttiin tietää kulmanopeuden muutos suhteessa aikaan. Kulmanopeuden yksikkö on radiaania sekunnissa, joten sen muutosnopeuskin laskettiin sekunneissa. Laskentalohkossa otettiin näyte, joka seuraavalla ohjelmakierroksella siirrettiin toiseen muistipaikkaan ja näistä näytteistä laskettiin erotus. Erotus jaettiin ohjelmakierron ajalla, jotta saatiin tulokseksi muutosnopeus sekuntia kohti. Kuviossa 30 on derivointilohkon ohjelmakoodi.

```

0001 #Derivative:=((#Input_New-#Input_Old)/#Cycle); //calculates difference between
    samples and divides it by cycletime. dn/dt
0002 #Input_Old:=#Input_New;
0003 #Input_New:=#Input;
0004

```

Kuvio 30 Derivointilohkon ohjelmakoodi

6.3.10 Massahitausmomentin laskenta

Massahitausmomentin kompensointia varten piti laskea massahitausmomentti. Koko kelan massahitausmomentti muodostui kahdesta osasta, kelan rakenteen pysyvistä ja nauhan aiheuttamasta muuttuvasta massahitausmomentista. Kelan metalliosien pysyvä massahitausmomentti oli syötettävä parametri, joka lisättiin laskettuun nauhan massahitausmomenttiin. Nauha muodosti kelan ympärille ontton lieriön muotoisen kappaleen, jonka massahitausmomentti voitiin laskea kaavalla 4.

$$J = \frac{1}{2}m(r^2 + R^2) \quad (4)$$

Kaavassa 4, m on lasikuitunauhan massa, r on kelan puolan säde ja R on kelan säde. Kuviossa 31 on massahitausmomenttilaskennan ohjelmakoodi.

```
0001 #TotalInertia:=0.5*#Mass*((#MinRadius*#MinRadius)+(#Radius*#Radius))+#StaticInertia;
0002 //calculating webs inertia and added to empty reels inertia
```

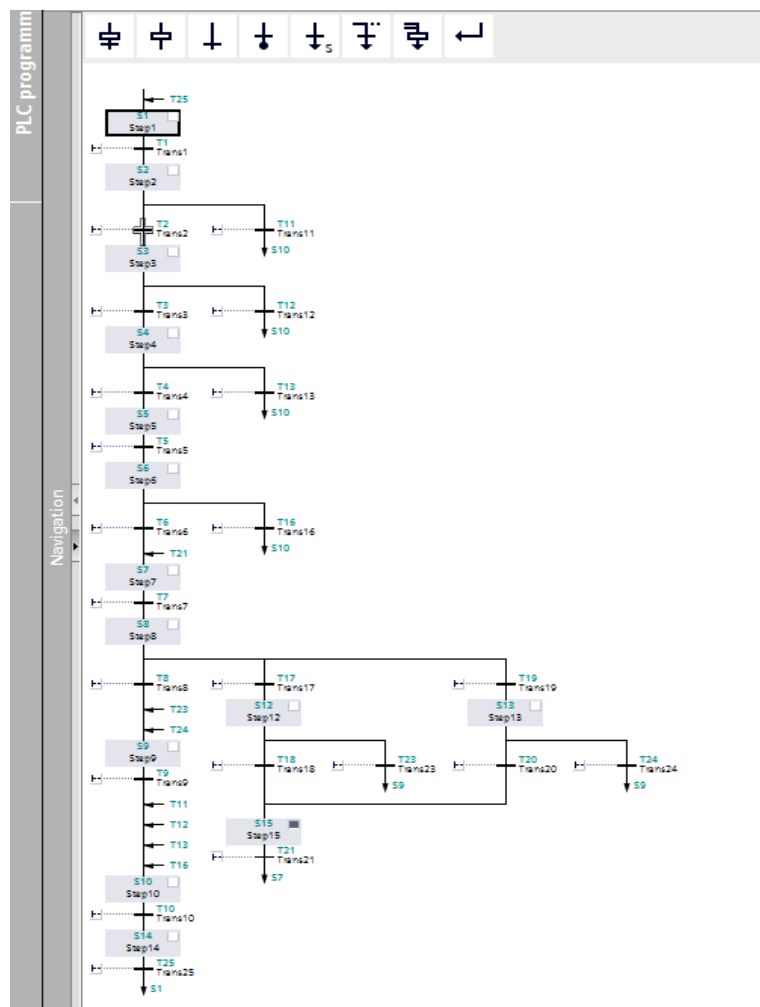
Kuvio 31 Massahitausmomenttilohkon ohjelmakoodi

6.4 Kelausohjelma

Kelaimen päätoimintona on nauhan kelaaminen aukikelaimelta kiinnikelaimelle. Nauhan kireyttä pyritään pitämään mahdollisimman tasaisesti asetusasteessa kiihdytyksen, kelauksen ja jarrituksen aikana.

Kelausohjelma on tehty kahteen eri lohkoon. Mittaukset, indikoinnit, toimilaitteiden ohjaukset ja niihin liittyvät toiminnot on tehty yhteen lohkoon. Näitä lohkon osia taas hallitaan GRAPH-editorilla toteutetulla sekvenssillä. Kuviossa 32 on valmis kelaussekvenssi. Sekvenssin stepeillä hallitaan ohjauksia ja trans-osilla määritellään millä ehdoilla siirrytään seuraavaan steppiin.

Sekvensseillä saavutetaan helposti hallittava ohjelmakokonaisuus. Sekvenssin peruskierrossa käydään kuvion 32 vasemmassa laidassa olevat stepit läpi. Mikäli jokin laite menee vikatilaan tai painetaan stoppia, voidaan kyseisellä ohjelmarakenteella määritellä kohdasta riippuen mihin steppiin ohjelma siirtyy. Ilman sekvenssejä on hankala määritellä pysäytyskäskyn tullessa, mikä laite tekee mitään ja kuinka se tuodaan hallitusti alas muiden laitteiden kanssa. Liitteessä 1 on kelaussekvenssin kuvaus ja selitykset askelista ja siirtymäehdoista.



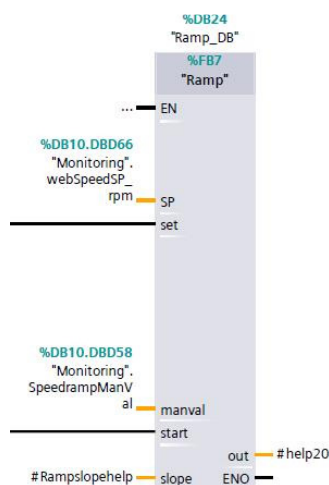
Kuvio 32 GRAPH-editorilla toteutettu kelaussekvenssi

6.4.1 Kiristys

Kelaussekvenssi alkaa nauhan kiristyksellä. Nauhan kiristys tapahtuu kahdessa vaiheessa. Ensimmäisessä vaiheessa kaikkia taajuusmuuttajia säädetään nopeusohjeella ja niille annetaan nopeusohje nopeuserolla. Aukikelaimelle annetaan nolla-nopeusohje. Seuraavalle taajuusmuuttajalle annetaan nopeuseroparametrin suuruinen nopeusohje ja sitä seuraavalle kaksi kertaa nopeuseroparametrin suuruinen nopeusohje. Kiinnikelaimelle annetaan kolme kertaa nopeuseroparametrin suuruinen nopeusohje. Kiristyksen tapahtuessa toiseen suuntaan nopeusohjeet annetaan negatiivisina.

Taajuusmuuttajia ajetaan nopeusohjeella siihen asti, että voimanmittaus saavuttaa asetetun raja-arvon. Nauhan liian kuormituksen välttämiseksi kiristyksen aikana

Kiihdytyksen ja hidastuksen pitää tapahtua hallitusti. Tämä saadaan aikaiseksi rampilla. Rampin tarkoituksena on rajoittaa taajuusmuuttajien saaman nopeusohjeen muutosnopeutta. Kuviossa 34 on kelainsovelluksessa käytetty ramppi. Rampissa käytetty muutosnopeus (slope) määrittelee kuinka paljon lähtö voi muuttua yhden ohjelmakierron aikana. Normaaliajossa rampin nousunopeudeksi määriteltiin 0-1000rpm 20 sekunnissa. Ohjelmakierron ollessa 20 ms, muutosnopeudeksi tuli yksi. Nopean pysäytyksen aikaansaamiseksi rampin muutosnopeudeksi asetetaan 10.



Kuvio 34 Ramppilohko

Pelkän lineaarisen rampin käyttäminen nopeusohjeen muodostuksessa voi aiheuttaa turhia rasitteita koneistolle. Käyttämällä s-ramppia voidaan hidastaa rampin alku- ja loppupään kiihtyvyyden muutosta. Moottoreiden on myös helpompi saavuttaa asetettu nopeus, kun asetuspistettä lähestyttäessä muutosnopeus hidastuu. Työssä käytettiin kahdenkymmenen näytteen liukuvaa keskiarvoa s-rampin muodostamiseen. Kuviossa 35 on taajuusmuuttajan saama nopeusohje sinisellä ja toteutunut nopeus on punaisella.



Kuvio 35 Taajuusmuuttajan saama nopeusohje ja toteutunut nopeus

6.4.3 Momenttiohjeen muodostus

Nauhan kireyttä hallitaan kiinni- ja aukikelaimen momenttiohjeella. Taajuusmuuttajalle tuleva momenttiohje muodostetaan useasta osasta. Ensimmäinen momenttiohjeen osa lasketaan kireysohjeesta ja kelan säteestä. Ohje lasketaan käyttämällä voiman momentti kaavaa (6). /3/

$$M = F \cdot r \quad (6)$$

Lasku antaa tuloksen Newton-metreinä ja taajuusmuuttajalle ohje syötetään promilleina. Laskusta saatu momenttiohje skaalataan nimellismomentin avulla promilleiksi. Ohjelman momenttiohjeen ensimmäisen osan koodi on kuviossa 36.

```
0001 //calculatin torquereference
0002 #TorqRef:=((#WebTensSP*#ReelRadius)/"Parameters".NominalTorq)*1000;
0003
```

Kuvio 36 Momenttiohjeen ensimmäisen osan ohjelmakoodi

Tällä momenttiohjeella kelain saa aikaiseksi vakionopeudella tasaisena pysyvän kireyden. Kireys on kuitenkin jakaantunut epätasaisesti. Kiinnikelaimen puolella on liian pieni kireys ja aukikelaimen puolella on liian suuri kireys. Tätä tilannetta pyritään tasoittamaan kompensoimalla kitkamomenttia.

Aukikelain saa nauhankireyden aikaiseksi jarruttamalla momenttiohjeen mukaisella voimalla. Kiinnikelain taas saa nauhan kireyden aikaiseksi kiihdyttämällä momenttiohjeen mukaisella momentilla. Kitkamomentti pyrkii jarruttamaan mo-

lempia keloja. Aukikelaimen momentti on jarruttava ja kun siihen lisätään vielä kitkamomentti, sen kokonaismomentti aiheuttaa liian suuren nauhan kireyden. Kiinnikelaimen momentti pyrkii kiihdyttämään kela ja kun siihen lisätään jarruttava kitkamomentti, sen kokonaismomentti jää liian pieneksi.

Kitkan kompensoinnilla pyritään poistamaan jarruttavan kitkamomentin vaikutus. Liukulaakerin kitkamomentti määritellään kaavan seitsemän mukaan. /3/

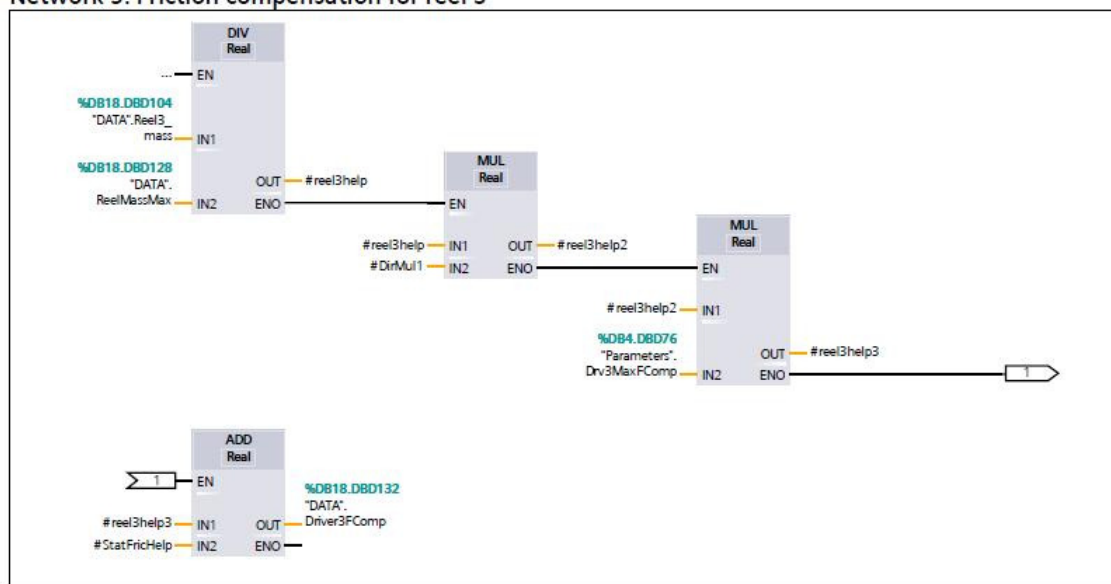
$$M = \mu * K * r \quad (7)$$

Kaavassa r on laakerin säde, K on kuorma ja μ on tappikitkan kerroin. Tappikitkan kerroin riippuu nopeudesta, voitelusta, aineista, kuormasta ja välyksestä. Kelaimessa kitkavoiman kompensoinnissa otettiin huomioon pelkkä massan aiheuttaman kuormituksen muutos. Kitkavoimien kompensointi toteutettiin kertomalla suurin kitkamomentin kompensointilisä kelan suhteellisella massalla. Lasku tehtiin kaavan 8 mukaan.

$$M = \frac{m}{m_{max}} * M_{fcomp} \quad (8)$$

Maksimi kompensointimomenttilisä määriteltiin koeajamalla kelainta eri suuntiin, eri kireysohjeilla ja eri nopeuksilla. Molemmille keloille määriteltiin omat kelaussuunnasta riippuvat kompensointimomentit. Tällä saavutettiin tähän käyttöön riittävä tarkkuus. Nopeuden pysyessä tasaisena nauhan kireys saatiin pysymään lähellä asetusta ja kireys jakaantui tasaisemmin. Kuormitusta määritettäessä ei otettu huomioon nauhan kireyden vaikutusta. Nopeuden ollessa alle yksi, kitkan kompensointiin lisättiin myös lepokitkan kompensointi. Kuviossa 37 on esimerkki kitkan kompensointilohkosta.

Network 5: Friction compensation for reel 3

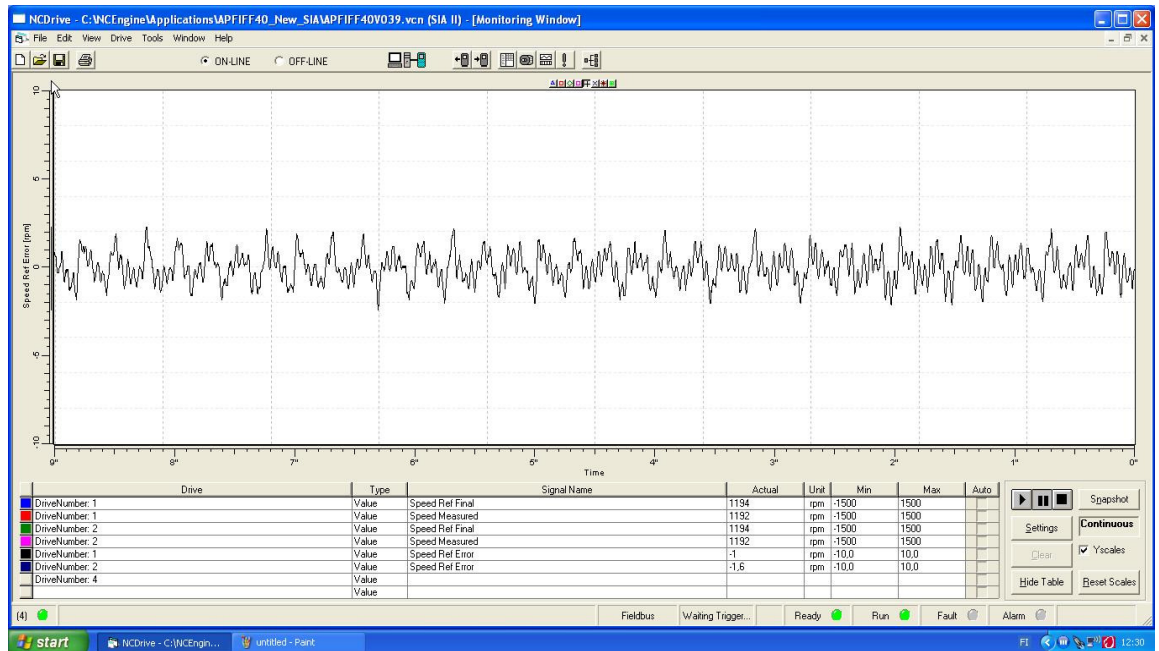
**Kuvio 37** Kitkan kompensoinnin ohjelmakoodia

Nopeuden pysyessä tasaisena nauhan kireyskin pysyy tasaisena. Kiihdytys tai jarrutustilanteissa kelojen massahitus pyrkii vastustamaan kulmanopeuden muutosta ja se aiheuttaa myös muutoksen nauhan kireyteen. Tätä voidaan kompensoida laskemalla kelojen tarvitsema momentin lisäys ja vähennys. Tarvittava momentin muutos voidaan laskea pyörimisen liikeyhtälöstä kaavan yhdeksän mukaan. /3/

$$\sum M = J\alpha \quad (9)$$

Kaavassa summamomentti vastaa kulmanopeuden muutokseen tarvittavaa kiihdyttävää momenttilisää. J tarkoittaa massahitusmomenttia ja α tarkoittaa kulmakiihtyvyyttä. Kelan massahitusmomentti J saadaan laskentalohkosta. Kulmakiihtyvyys α voidaan laskea nauhan nopeusohjeen derivaatan avulla. Kelan kulmakiihtyvyys lasketaan nauhan nopeusohjeesta, jotta kompensointi saataisiin toimimaan ajoissa.

Todellisen nopeudenkäyttämisen estää myös mitatun nopeuden vaihtelu. Kuviossa 38 on NCDrivella otettu kuva nopeussäädöllä olevan linjamoottorin nopeusvirheestä. Nopeuden vaihteluväli ei ole suuri, mutta sen derivaatta on suuri.



Kuvio 38 Linjamoottorin nopeusvirhe

Nopeuden derivaatta oli kannattava laskea nopeusohjeesta myös sen takia, että momentti lisää-/vähennysohje tulisi taajuusmuuttajalle heti, kun nopeuden muutos alkaa.

Kelojen nopeuden derivaatta voitiin laskea nopeusohjeesta halkaisijoiden suhteella. Kaavassa 10 on esitetty kaksi halkaisijaltaan erisuuria kela, joiden kehänopeuden derivaatta on yhtä suuri.

$$d_1 * \pi * \frac{dn_1}{dt} = d_2 * \pi * \frac{dn_2}{dt} \quad (10)$$

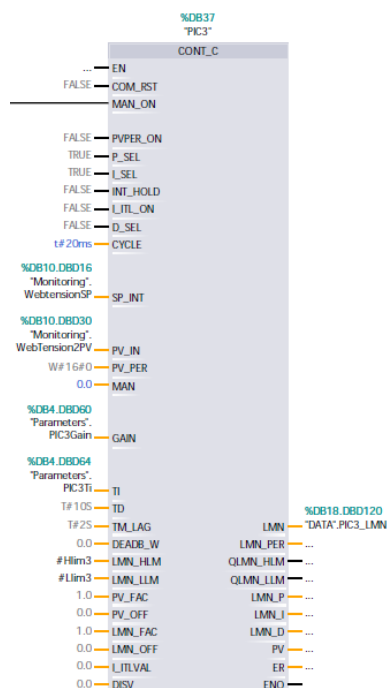
Kaavassa d_1 esittää linjatelan halkaisijaa ja dn_1/dt on linjanopeusohjeen derivaatta. d_2 on kelan halkaisija ja dn_2/dt kelan pyörimisnopeuden derivaatta. Kaavasta voidaan ratkaista kelan nopeuden derivaatta. Ratkaisu on esitetty kaavassa 11.

$$\frac{dn_2}{dt} = \frac{d_1}{d_2} * \frac{dn_1}{dt} \quad (11)$$

Massahitausmomenttia kompensoiva kiihdyttävä momenttiohje saatiin pyörimisen liikeyhtälön mukaisesti kertomalla kelan massahitausmomentti kulmakiihtyvyydellä (suhteutetulla nopeusohjeen derivaatalla).

Massahitausmomentin kompensointiohje syötettiin taajuusmuuttajan torque step-toimintoon. Torque step:iin tuleva momenttiohje ei mene momenttiohjeen suodatimen läpi ja sen ansiosta sillä saavutetaan nopeampi vaste. Näillä momenttiohjeen osilla nauhan kireyden keskiarvo saatiin pidettyä kohtalaisen vakaana ja lähellä asetusarvoa.

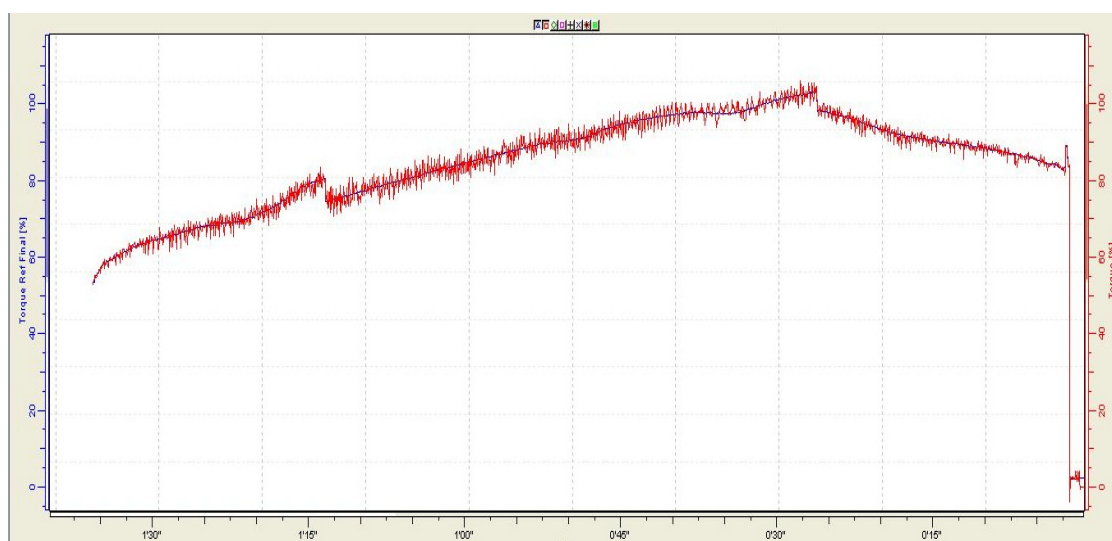
Kireyden pitämiseksi asetusarvossa käytettiin PI-säätäjää. Säätäjänä käytettiin Siemensin S7-300-sarjan omaa Continues controller-lohkoa. Lohko on kuviossa 39. TIA-Portalilla lohkon toiminnan voi määrittellä technology objects comissionin kohdasta. Kyseistä valikosta määriteltiin säätäjä toimimaan PI-säätäjänä. Säätäjän muut määrittelyt toteutettiin lohkon inputeilla. Säätäjälle tuotu nauhan kiireystieto oli liukuva keskiarvo mitatusta kireystiedosta, jotta säätäjä ei koettaisi korjata pientä heiluntaa vaan pitäisi keskiarvon vakaana. Säätäjälohko oli käytössä vain automaattikelauksen aikana. Kelaimen ollessa muussa tilassa, säätäjä asetettiin manuaalitilaan ja lähtö nolnaan.



Kuvio 39 Continues controller-lohko

Lopullinen momenttiohje muodostettiin laskemalla kaikki momenttiohjeet yhteen. Kuviossa 40 on esitetty taajuusmuuttajan saama lopullinen momenttiohje sinisellä

ja taajuusmuuttajan laskema toteutunut momentti punaisella. Kuvan alkuosassa näkyy momenttiohjeen notkahdus. Tämä johtuu siitä, kun kelain saavuttaa asetetun ratanopeuden, eikä massahitausmomenttia tarvitse enää kompensoida lisämomenttiohjeella. Kuvan loppuosassa momenttiohje tipahtaa taas äkillisesti. Tässä vaiheessa kelain alkaa hidastaa nopeutta ja massahitausmomenttia kompensoidaan momenttiohjeen vähennyksellä. Kuvan lopussa olevasta piikistä voidaan päätellä, että massahitausmomentin kompensointi on loppunut liian aikaisin.



Kuvio 40 Kelauksen aikainen momenttiohje ja toteutunut momentti

6.4.4 Kelaimen pysäytys

Kelaimen pysäytys riippuu missä vaiheessa sekvenssiä, ja mistä syystä pysäytyskäsky tulee. Mikäli kelaussekvenssi ei ole päässyt kelaukseen asti, kaikki pysäytyskäskyt asettavat pysäytyksen aloittavan askeleen aktiiviseksi. Tässä tapauksessa taajuusmuuttajille annetaan vain stop-käsky. Mikäli kelaus on käynnissä ja annetaan pysäytyskäsky, ajetaan nopeus rampilla hitaasti alas. Linjan nopeuden ollessa nollassa, taajuusmuuttajille annetaan stop-käsky. Prosessi ajetaan samalla tavalla alas, kun tulee moottorin yllilämpöindikointi, ympäristön yllilämpöindikointi tai taajuusmuuttajan varoitusindikointi. Quickstop-käsky ja taajuusmuuttajan vikaindikointi ajaa prosessin alas samalla tavalla, mutta asettaa rampin nousunopeuden kymmenkertaiseksi. Tässä tapauksessa prosessi saadaan pysähtymään hallitusti 2-3 sekunnissa, riippuen linjanopeudesta.

Oven auetessa laite pyritään pysäyttämään mahdollisimman nopeasti. Pysäytys toteutetaan asettamalla kaikki taajuusmuuttajat nopeusohjeelle ja niille annetaan nollanopeusohje. Pysäytyksen nopeutta ja nauhaan kohdistuvaa räsistystä pyritään hallitsemaan pysäytyksen aikaisilla momenttirajoilla. Logiikan ohjatessa jarruja, myös jarrut osallistuvat pysäytykseen ovirajan auetessa.

6.4.5 Loop-toiminto

Kelaimelle haluttiin niin sanottu loop-toiminto, missä kelain kelaau nauhan päästä päähän ja vaihtaa automaattisesti suuntaa kunnes asetettu kelausmäärä täyttyy. Loop-toiminto toteutettiin tekemällä kelaussekvenssiin vaihtoehtoinen haara pysäytykselle. Loop-toiminnon ollessa aktiivinen, sekvenssi ei siirtynyt automaattisen kelauksen pysäytyksen jälkeen sammuttamaan taajuusmuuttajia, vaan siirtyi rinnakkaiseen loop-haaraan. Haarassa tarkastettiin kelaussuuntaa ja toteutuneiden kelausten määrä. Määrän ollessa täysi siirryttiin takaisin päähaaraan sammuttamaan taajuusmuuttajia. Muussa tapauksessa vaihdettiin kelaussuuntaa, lisättiin kelauslaskuriin yksi ja siirrettiin aiemmin käytetty nauhan nopeusohje uudeksi nopeusohjeeksi. Näiden toimintojen jälkeen hypättiin takaisin normaaliin kelausaskeliseen. Liitteessä 1 on näytetty kaaviokuva askelista ja siirtymistä. Liitteessä olevissa taulukoissa on selitetty askeleen ohjaamat toiminnot ja siirtymäehdot.

6.4.6 Kelauksen aikaiset rajoitukset

Kelain on tarkoitettu koulutus- ja esittelykäyttöön. Laitetta tulisi siis käyttää ihmiset, joilla ei ole kunnollista perehdytystä laitteen käyttöön. Tämän vuoksi käyttöä piti rajoittaa mahdollisten väärinkäytösten vuoksi. Mahdollisia väärinkäyttötilanteita oli hankala määritellä etukäteen. Suurin osa näistä tilanteista ilmeni koekäytöissä, jonka jälkeen kyseinen väärinkäyttöön johtanut toiminto estettiin kyseisessä tilanteessa.

Kelauksen aikaisia rajoituksia oli ovirajan ohituksen esto, local/remote -tilan vaihto kesken ajon, jarrujen ohjauspaikan valinta, käytettävän kenttäväylän valinta, ryömintä, kiristyssekvenssi ja manuaalitilaan vaihto. Manuaaliohjaus, kiristysse-

kvenssi, jarrujen ohjauspaikan ja kenttäväylän valinta oli sijoitettu salasanalla suojatulle sivulle. Lisäksi niiden käytöllä oli asetettu ehto, että kelain on pysähdyksissä ja local -tilassa.

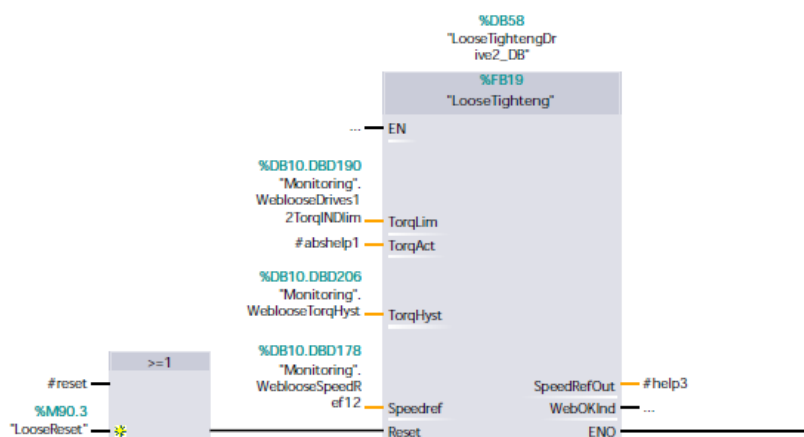
Ovirajan ohituskytkimen kääntäminen kesken kelauksen asetti nopeusohjeen nolnaan. Kelain ajettiin rampilla nollanopeuteen, eikä nopeusohjetta voinut muuttaa ennen kuin ohituskytkimen palauttaa normaaliasentoon.

Kelaussekvenssin aloituksella oli ehtona, että laite on Remote-tilassa. Mikäli kyt-kin vaihdettiin kelauksen aikana local-tilaan, laite alkoi pysäyttää prosessia. Tilaindikointi säilyi remotena siihen asti, että prosessi ja taajuusmuuttajat oli pysähdyksissä. Vasta tämän jälkeen indikointi vaihtui local-tilaan.

6.5 Kiristyssekvenssi

Laitteeseen haluttiin erillinen kiristystoiminto, jolla pystytään kiristämään nauha automaattisesti, mikäli siinä on paljon löysää. Tämän tapaisia tilanteita saattaa syntyä, jos taajuusmuuttajaan tulee vika ajon aikana.

Kiristyssekvenssi toteuttaa kiristyksen kahdessa vaiheessa. Ensimmäisessä vaiheessa kiristetään kelaimen linjaosuus, jonka jälkeen kiristetään kelojen osuus. Kiristyssekvenssiä varten tehtiin kuvion 41 mukainen lohko.



Kuvio 41 Kiristyslohko

Kiristyssekvenssissä taajuusmuuttajia ohjataan nopeusohjeella. Lohkoon annetaan kiristuksen aikana käytettävä nopeusohje. Nauhan kireyden indikoinnille asetetaan momentti- tai kireysraja sekä hystereesi. Lohko antaa taajuusmuuttajalle nopeusohjeen, kunnes mitattu momentti tai nauhan kireys saavuttaa asetetun rajan. Tämän jälkeen lohko antaa indikoinnin siitä, että nauhan kireys on kunnossa.

Linjamootoreita pyöritetään vastakkaisiin suuntiin, kunnes niiden momentti ylittää asetetun rajan. Tämän jälkeen keloja pyöritetään vastakkaisiin suuntiin, kunnes voima-antureilla mitattu nauhan kireys vastaa asetettua rajaa ja lohkot antavat indikoinnin. Kun kaikilta kiristyslohkoilta on saatu indikointi, sekvenssi sammuttaa taajuusmuuttajat. Kiristyssekvenssin aikana käytetään alemmaa momenttirajaa, jottei nauhalle aiheutettaisi tarpeetonta räsitusta nauhan ollessa kierteellä tai solmussa.

6.6 Ryömintä

Ryömintätilaa voidaan käyttää, kun kelain on local-tilassa. Ryömintätilassa kelaainta voidaan ajaa hitaalla nopeudella valittuun suuntaan. Suunta valitaan demolaitteen kyljessä olevalla nokkakytkimellä. Ryömintää voidaan käyttää myös oven ollessa auki.

Ryömintätoiminto on toteutettu sekvenssillä. Ensin ohjelma asettaa taajuusmuuttajan tyyppilohkot auto-tilaan. Tämän jälkeen kelojen taajuusmuuttajat asetetaan

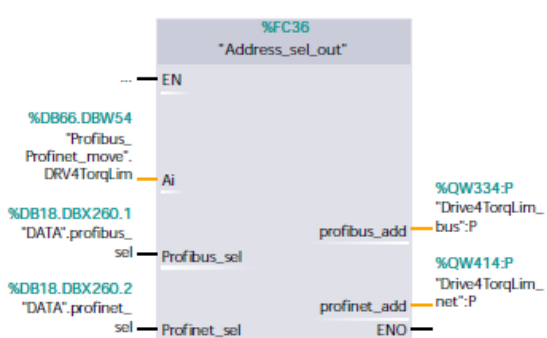
momenttiohjaukselle ja linjan taajuusmuuttajat asetetaan nopeusohjaukselle. Linjamooottoreille annetaan nopeusohje positiivisena tai negatiivisena, riippuen valitusta ryömintäsuunnasta.

Aukikelaimen halkaisijan ollessa alle raja-arvon, kelain ei aloita ryömintää. Ryömintä pysähtyy, kun ohjauskytkin palautetaan alkuasentoon tai aukikelaimen halkaisija laskee alle raja-arvon. Ryöminnässä käytettävät nopeus ja momenttiohjeet ovat parametroitu logiikassa, eikä niitä voida muuttaa käyttöliittymästä.

6.7 Kenttäväylän valinta

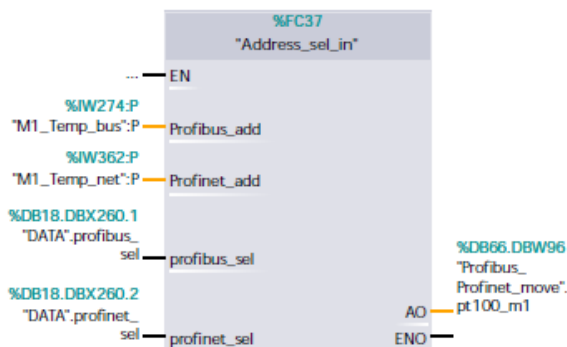
Laitteen toiminnallisuutta määriteltäessä päädyttiin siihen, että laitetta voitaisiin ohjata valinnaisesti PROFIBUS DP:llä tai PROFINET I/O:lla. Väylän vaihdon tuli onnistua käyttöliittymästä. Kenttäväylän vaihto-ominaisuuden vuoksi logiikassa oli kahdet periferiaosoitteet kaikille NXP-taajuusmuuttajille. Hardwareasetuksissa määritellyt kenttäväylälaitteet viittaavat taajuusmuuttajien väyläliikennöintikortteihin, eikä itse laitteisiin. Tämän vuoksi laitteelle on kahdet eri periferiaosoitteet riippuen käytettävästä väylästä.

Vaihdettaessa liikennöintiväylää, ohjelmasta piti muuttaa periferiaosoitteet vastaamaan käytettävän väylän liikennöintikortin osoitteita. Tämä toteutettiin kahdella erillisellä lohkoilla. Toinen lohko oli tuloille ja toinen lähdöille.



Kuvio 42 Lähtöosoitteen valintalohko

Kuviossa 42 on taajuusmuuttajalle kirjoitettavan datan osoitteenvalintalohko. Aikanavaan kirjoitetaan lähetettävä tieto. Profibus_sel ja Profinet_sel valitsee mihin periferia osoitteeseen Ai-kanava kirjoitetaan.



Kuvio 43 Tulo-osoitteen valintalohko

Kuviossa 43 on taajuusmuuttajalta luettavan datan osoitteenvalintalohko. Profibus_add kohtaan tulee PROFIBUS-väylän osoite ja Profinet_add kohtaan PROFINET-väylän osoite. Ao:sta saadaan ulos valitun osoitteen sisältämä data. Kaikki taajuusmuuttajalle menevä tieto meni kuvion 42 mukaisen lohkon kautta ja kaikki taajuusmuuttajalta tuleva tieto tuli kuvion 43 mukaisen lohkon kautta.

Väylän kautta kulki jatkuvasti dataa taajuusmuuttajille ja väylää vaihdettaessa PROFIBUS-väylästä PROFINET-väylään, tämä aiheutti väylävirian. Väyläviria ei tullut vikailmoitusta, vaan taajuusmuuttaja jäädytti logiikalta tulevat prosessidatat viimeisimpiin arvoihin mitkä olivat ennen vikaa. Kyseisen virian ollessa päällä taajuusmuuttaja tottelee Control Word:a ja nopeus-referenssiä. SW, todellinen nopeus ja taajuusmuuttajan lähettämät prosessidatat kulkevat normaalisti.

Väylän arvojen jäädyttämisen pystyi estämään taajuusmuuttajan parametrilla P2.10.3 Control Options. Parametriin piti asettaa desimaaliluku 4096, joka vastaa bittiä 12. Parametriin pystyi asettamaan vain yhden bitin kerrallaan. Oletuksena parametrin arvo oli 64, mikä vaikutti open loop-toimintaan. Kelaimessa taajuusmuuttajia käytettiin closed loop-tilassa, joten parametrilla ei ollut merkitystä tässä tilanteessa. Tämä toiminto ei kuitenkaan estänyt väylävirian ilmaantumista vaan se jäi päälle. Viriasta ei kuitenkaan saanut ilmoitusta SW:ta eikä laitteesta.

6.8 Vika- ja varoitussanat

Taajuusmuuttajalla on vika- ja varoitustilanteiden analysointia varten kaksi viikasanaa ja yksi varoitussana. Yksi sana koostuu 16 bitistä, joista yksittäiset bitit

indikoivat tiettyä vikaa. Bittien merkitykset on selitetty taajuusmuuttajan soveluksen manuaalissa. Kuviossa 44 on manuaalista otettu taulukko vikasanan bittien selityksistä.

	Fault Word 1 ID1172	
	Fault	Comment
b0	Over Current or IGBT	F1, F31, F41
b1	Over Voltage	F2
b2	Under Voltage	F9
b3	Motor Stalled	F15
b4	Earth Fault	F3
b5	Motor Under Load	F17
b6	Drive over temperature	F14
b7	Over Temperature	F16, F56, F29
b8	Input Phase	F10
b9	Brake resistor over temperature	F42 (Not implemented)
b10	Device Changed	F37, F38, F39, F40, F44, F45 (Not implemented)
b11	Keypad or PCControl	F52
b12	FielBus	F53
b13	SystemBus	F59
b14	Slot	F54
b15	4 mA	F50

Kuvio 44 Taajuusmuuttajan vikasanan selitykset

Vikasanat käsiteltiin logiikassa, josta ne lähetettiin käyttöliittymään. Vikasanat ja varoitussana tuotiin lohkoon, jossa niiden bitit luettiin yksitellen ja siirrettiin datablokkiin vastaavaan kohtaan. Kuviossa 45 on osa vikasanan datablokista.

Name	Data type	Offset	Default value	Retain	Accessible from HMI	Visible in HMI	Set-point	Comment
▼ Input								
FW1Add	Word	0.0	16#0	Set in IDB	True	True	False	Faultword1 address
FW2Add	Word	2.0	16#0	Set in IDB	True	True	False	Faultword2 address
WWAdd	Word	4.0	16#0	Set in IDB	True	True	False	Warningword address
reset	Bool	6.0	false	Set in IDB	True	True	False	reset faultwords
Output								
InOut								
▼ Static								
FW1_b0	Bool	8.0	false	Set in IDB	True	True	False	Over Current or IGBT
FW1_b1	Bool	8.1	false	Set in IDB	True	True	False	Over Voltage
FW1_b2	Bool	8.2	false	Set in IDB	True	True	False	Under Voltage
FW1_b3	Bool	8.3	false	Set in IDB	True	True	False	Motor Stalled
FW1_b4	Bool	8.4	false	Set in IDB	True	True	False	Earth Fault
FW1_b5	Bool	8.5	false	Set in IDB	True	True	False	Motor Under Load
FW1_b6	Bool	8.6	false	Set in IDB	True	True	False	Drive Over Temperature
FW1_b7	Bool	8.7	false	Set in IDB	True	True	False	Over Temperature
FW1_b8	Bool	9.0	false	Set in IDB	True	True	False	Input Phase
FW1_b9	Bool	9.1	false	Set in IDB	True	True	False	Brake Resistor Over Temperature
FW1_b10	Bool	9.2	false	Set in IDB	True	True	False	Device Changed
FW1_b11	Bool	9.3	false	Set in IDB	True	True	False	Keypad or PCControl
FW1_b12	Bool	9.4	false	Set in IDB	True	True	False	Fieldbus
FW1_b13	Bool	9.5	false	Set in IDB	True	True	False	SystemBus
FW1_b14	Bool	9.6	false	Set in IDB	True	True	False	Slot
FW1_b15	Bool	9.7	false	Set in IDB	True	True	False	4 mA
FW2_b0	Bool	10.0	false	Set in IDB	True	True	False	Output phase
FW2_b1	Bool	10.1	false	Set in IDB	True	True	False	Charge Switch
FW2_b2	Bool	10.2	false	Set in IDB	True	True	False	Encoder
FW2_b3	Bool	10.3	false	Set in IDB	True	True	False	Inverter
FW2_b4	Bool	10.4	false	Set in IDB	True	True	False	

Kuvio 45 Vikasana-DB

Kun vikasanat oli purettu biteiksi, oli helpompi tulkita aktiivisia vikoja. Niiden tulkitsemista varten tehtiin lohko, joka kokosi vikasanat yhdeksi string-muuttujaksi. Kuviossa 46 on osa vikasanan muodostuslohkosta.


```

0001
0002 //writing faults and warning to Faultword
0003 IF #FWB0 = true THEN
0004   #Faultword:=CONCAT(IN1:= #Faultword, IN2:='over current OR IGBT');
0005   END_IF;
0006 IF #FBB1 = true THEN
0007   #Faultword:=CONCAT(IN1:=#Faultword, IN2:='Over voltage');
0008   END_IF;
0009 IF #FWB2 = true THEN
0010   #Faultword:=CONCAT(IN1:=#Faultword, IN2:='Under voltage');
0011   END_IF;
0012 IF #FWB3 = true THEN
0013   #Faultword:=CONCAT(IN1:=#Faultword, IN2:='Motor Stalled');
0014   END_IF;

```

Kuvio 46 Vikasanan tulkintalohkon ohjelmakoodia

Lohko tarkasti aktiiviset vikabitit ja lisäsi sitä vastaavan vikatekstin vikasanaan. Vikaindikoinnin tullessa lohkoa pidettiin päällä yhden ohjelmakierron ajan, jolloin se kirjoitti vikasanan vain kerran. Kun vika nollattiin, se muutti samalla vikasanan tyhjäksi. Lohkon tarkoituksena oli tuoda taajuusmuuttajan kaikki viat yhdellä muuttujalla käyttöliittymään valmiiksi luettavassa muodossa. Logiikan ja käyttöliittymän välinen liikennöinti esti kyseisen vikasanan käytön. Stringmuuttujan tulkinta käyttöliittymän puolella ei onnistunut. Muuttujaan tuli ylimääräisiä merkkejä. Tämän vuoksi tehtiin myös toinen lohko. Lohkossa jokaista vikabittiä vastasi tietty lukuarvo, joka kirjoitettiin lähtöön. Tässä oli ongelmana se, että se pystyi tuomaan käyttöliittymään vain yhden vian kerrallaan. Esimerkki lohkoista on kuviossa 47.

```

0002 IF #FWB0 = TRUE THEN
0003   #FaultINT:=1;
0004 ELSIF #FWB1 = true THEN
0005   #FaultINT:=2;
0006 ELSIF #FWB2 = true THEN
0007   #FaultINT:=3;
0008 ELSIF #FWB3 = true THEN
0009   #FaultINT:=4;
0010 ELSIF #FWB4 = true THEN
0011   #FaultINT:=5;
0012 ELSIF #FWB5 = true THEN
0013   #FaultINT:=6;
0014 ELSIF #FWB6 = true THEN
0015   #FaultINT:=7;
0016 ELSIF #FWB7 = true THEN
0017   #FaultINT:=8;
0018 ELSIF #FWB8 = true THEN
0019   #FaultINT:=9;
0020 ELSIF #FWB9 = true THEN
0021   #FaultINT:=10;
0022 ELSIF #FWB10 = true THEN

```

Kuvio 47 Vikasanan tulkintalohkon ohjelmakoodia

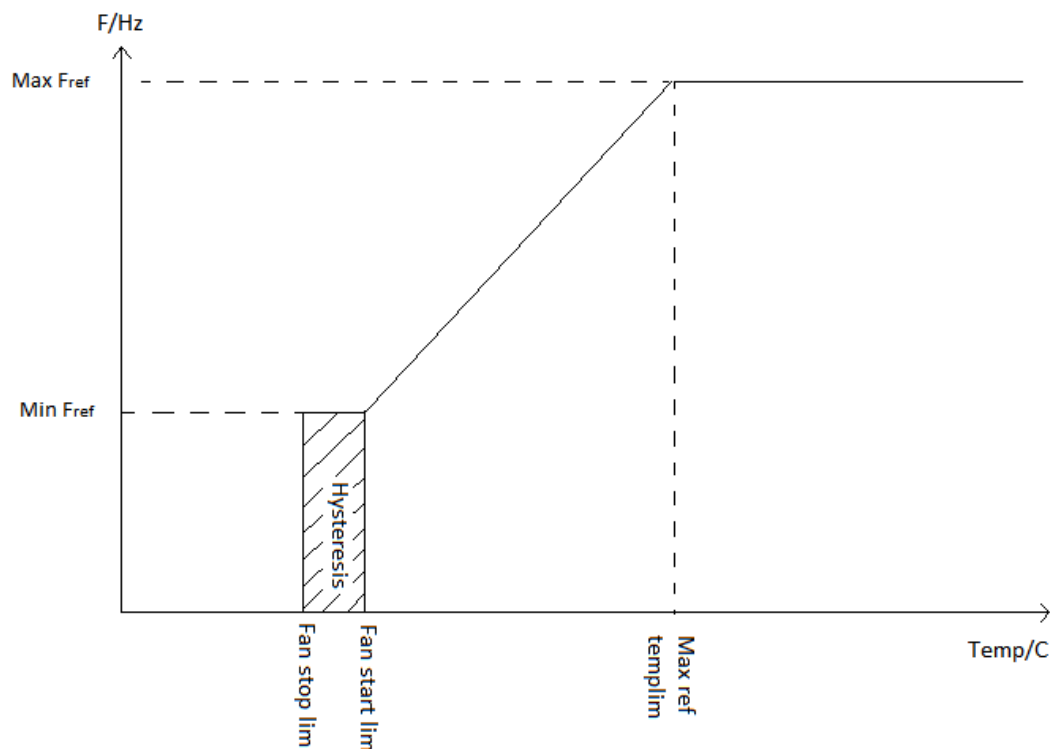
Lohkojen tarkoituksena oli tuoda vikatieto käyttöliittymään mahdollisimman vähällä muuttajamäärällä, sillä käyttöliittymän muuttujien määrä oli rajoitettu 64:n.

6.9 Prosessi- ja taajuusmuuttajien tilatieto

Käyttöliittymään tuotiin myös tilatietoja prosessista. Tilatiedot indikoivat yksinkertaisia tietoja, kuten kelaussekvenssin vaihetta. Tilasana tuotiin käyttöliittymään integer-muuttujana, jonka arvo vastasi tiettyä tilaa. Tällä tavoin käyttöliittymään pystyttiin tuomaan vain yksi tilatieto kerrallaan. Tilatiedot piti priorisoida logiikkaohjelmassa, jolla taattiin oleellisen viestin näkyminen käyttöliittymässä. Taajuusmuuttajien tilatiedot tuotiin myös integer-muuttujana käyttöliittymään. Taajuusmuuttajalta tuodut tilatiedot olivat ready, stop, run, fault ja warning.

6.10 Lämpötilan säätö

Kelaimessa oli viisi lämpötilanmittausta. Kaikissa neljässä moottorissa oli rungon lämpötilanmittaus ja kelainlaitteen sisälämpötilaa mitattiin kaapin yläosasta. Moottoreiden jäähdytys oli toteutettu erillisillä tuulettimilla, joita ohjattiin yhdellä taajuusmuuttajalla. Taajuusmuuttajien moottorin lämpötilanlaskenta ja suojaustoiminto otettiin pois käytöstä. Moottoreiden lämpötilan säädöstä ja valvonnasta vastasi ainoastaan logiikka. Moottoreiden tuuletus toteutettiin lämpötilaan verrannollisella nopeusohjeella. Kuviossa 48 on esitetty tuulettimien taajuusohje lämpötilan funktiona.



Kuvio 48 Tuulettimien nopeusohje

Tuulettimille parametroidiin minimilähtötaajuus ja lämpötila, jolloin tuulettimet käynnistetään. Tuulettimet pysähtyivät, kun lämpötila laski alle hystereesin. Tuulettimien ohjaukseen parametroidiin myös lämpötila, millä lähtötaajuus olisi maksimissa. Tuulettimien taajuusohje laskettiin korkeimman mitatun moottorilämpötilan mukaan.

Tuulettimien taajuusohje toteutettiin tällä tavoin, jotta laitteesta saataisiin mahdollisimman hiljainen. Mikäli tuulettimia ohjattaisiin PI-säätäjällä, pienikin asetuspisteen ylitys voisi antaa täyden taajuusohjeen. Käytetyllä ohjauksella tuulettimet saavat täyden taajuusohjeen vasta, kun lämpötila ylittää parametroidun lämpötilarajan.

Moottoreiden- ja ympäristönlämpötilalle on asetettu yläraja milloin ne antavat yli- lämpöindikoinnin. Ylilämpöindikointi ajaa prosessin alas ja estää sen uudelleen käynnistämisen. Indikoinneissa käytetään hystereesiä ja ne poistuvat vasta, kun lämpötila on laskenut alle hystereesin.

6.11 Jarrujen ohjaus

Kelainlaitteessa olevissa keloissa oli solenoideilla ohjattavat mekaaniset jarrut. Solenoidien ohjausjännite tuli taajuusmuuttajan relekortilta. Alkuperäisessä suunnitelmassa jarrujen ohjaus toteutettiin taajuusmuuttajien sovelluksella. Taajuusmuuttajan saadessa käynnistyskäskyn, ne avasivat jarrut. Jarrut kytkettiin takaisin päälle, kun taajuusmuuttaja sai stop-käskyn.

Työn tilaaja halusi, että nauha pysyisi kireänä myös laitteen ollessa pysähtyneenä. Tämä ei onnistunut taajuusmuuttajien jarruohjauksella. Taajuusmuuttajan jarruohjaustoiminto aktivoitui, kun taajuusmuuttaja sai stop-käskyn. Logiikan ohjatessa kaikkea toimintaa taajuusmuuttajat saivat stop-käskyn vasta, kun moottorit olivat täysin pysähtyneet. Jarrujen mekaanisesta ja sähköisistä viiveistä johtuen kelat pääsivät liikkumaan ennen kuin jarrut ottivat kiinni kelaan. Logiikan avulla jarrut voitiin kytkeä päälle, kun moottorit pitivät nauhaa kireänä.

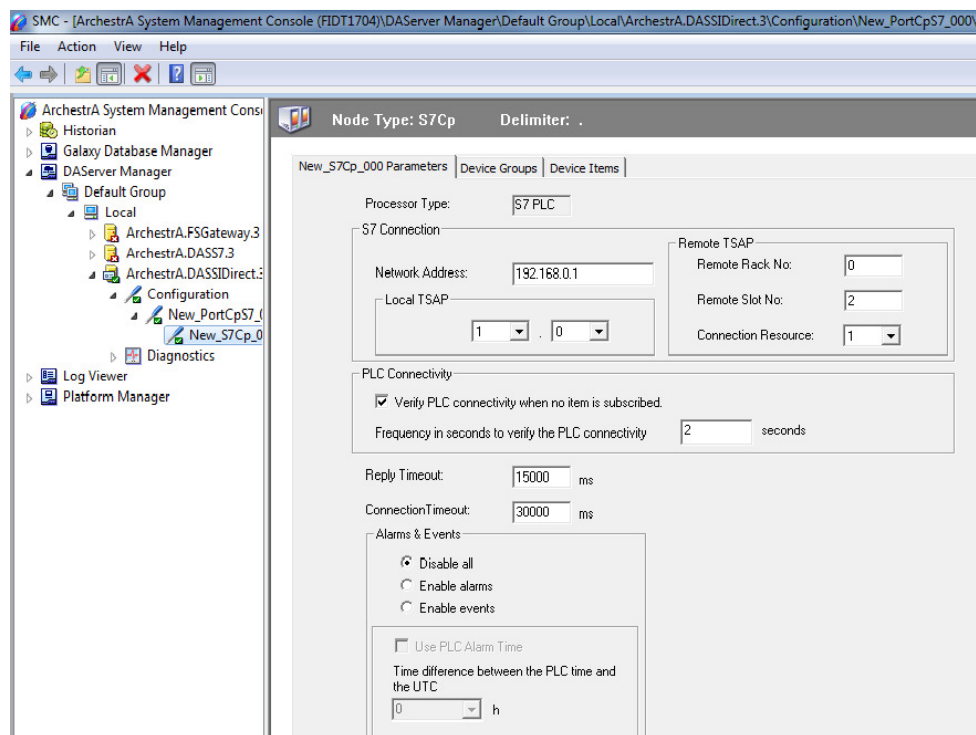
Tilaaja halusi myös käyttöliittymään mahdollisuuden valita, ohjaako jarruja logiikka vai taajuusmuuttajat. Logiikan ohjatessa jarruja, jarrut aukesivat taajuusmuuttajan saatua käynnistyskäskyn. Jarrujen kiinniohjauskäsky riippui, mitä kelaimella tehtiin. Automaattikelauksella jarrut saivat kiinniohjauskäskyn, kun prosessia ajettiin alas ja nauhan nopeus alitti asetetun rajan. Raja määriteltiin jarrujen toimintaviiveen mukaan. Taajuusmuuttajat saivat stop-käskyn vasta, kun kelat olivat pysähtyneet ja jarrut olivat päällä. Ryöminällä, yksittäisohjauksella ja kirstyssekvenssillä jarrut menivät kiinni, kun taajuusmuuttajalle annettiin stop-käsky.

7 KÄYTTÖLIITTYMÄ

Kelaimelle tehtiin myös käyttöliittymä, jolla prosessi voitiin hallita. Käyttöliittymä PC:n ja logiikan välistä yhteyttä hallittiin SMC-ohjelmalla. Käyttöliittymä ohjelmoitiin InTouchin Window Maker-ohjelmalla ja sitä käytettiin Window Viewer-ohjelmalla.

7.1 Yhteyden muodostaminen

Ensimmäinen vaihe käyttöliittymän tekemisessä oli yhteyden luominen logiikan ja käyttöliittymäohjelman välille. Tätä varten PC:lle asennettiin InTouch-ohjelmistopaketti, Windows SQL-server ja SIDirect DAServer-ohjelma. Yhteyttä varten logiikan hardware-asetuksista piti varmistaa liikennöintiresurssien asetukset. Logiikalla oli mahdollista muodostaa yhteys 12 eri laitteeseen. Niistä yksi oli varattu ohjelmointilaitteelle, yksi ohjauspaneelille ja loput S7-liikennöintiin. Käyttöliittymän ja logiikan välillä käytettiin S7-liikennöintiä, joten resurssiasetuksiin ei tarvinnut tehdä muutoksia. Käyttöliittymän päähän piti myös määritellä liikennöintiin tarvittavat asetukset. Kuviossa 49 nähdään käytetyt liikennöintiasetukset.



Kuvio 49 SMC-liikennöintiasetukset

Käyttöliittymän ja logiikan välistä liikennöintiä varten käytettiin Dasengine nimistä ohjelmaa. Dasengine toimi liikennöintirajapintana logiikan käyttämän S7-kommunikointiprotokollan ja käyttöliittymän SuiteLink-protokollan välillä.

Yhteysmäärittelyt tehtiin SMC ArchestrA System management consolella, jolla hallitaan logiikan ja käyttöliittymän välistä kommunikointia PC:n päästä. Yhteyttä varten luotiin uusi S7Cp node DASSIDirect3-kansion alle. Yhteyttä varten tarvittiin myös logiikan IP-osoite ja logiikan kommunikointiportin fyysinen sijainti laitteessa. Kommunikointiportin sijainti määriteltiin kohtaan Remote TSAP. Logiikan kommunikointiportti sijaitsi rackin numero nolla slotissa kaksi. Device groups-välilehdessä määriteltiin liikennöintiin tarvittava Topic ja tietojen päivitysnopeus. Käytetty Topic oli winder ja päivitysnopeudeksi asetettiin 1000ms.

SMC-asetukset olivat valmiit ja voitiin avata InTouch Window Maker-ohjelma, jolla itse käyttöliittymäsovellutus toteutettiin. Window Makerissa määriteltiin uudelle projektille Access name. Access name sisältää tiedot ohjelman nimestä, käytettävästä Topic-nimestä ja protokollasta. Topicin piti olla sama mikä määriteltiin

S7Cp node-kohtaan. Ohjelma oli DASSIDirect ja protokollana käytettiin Suitelinkiä.

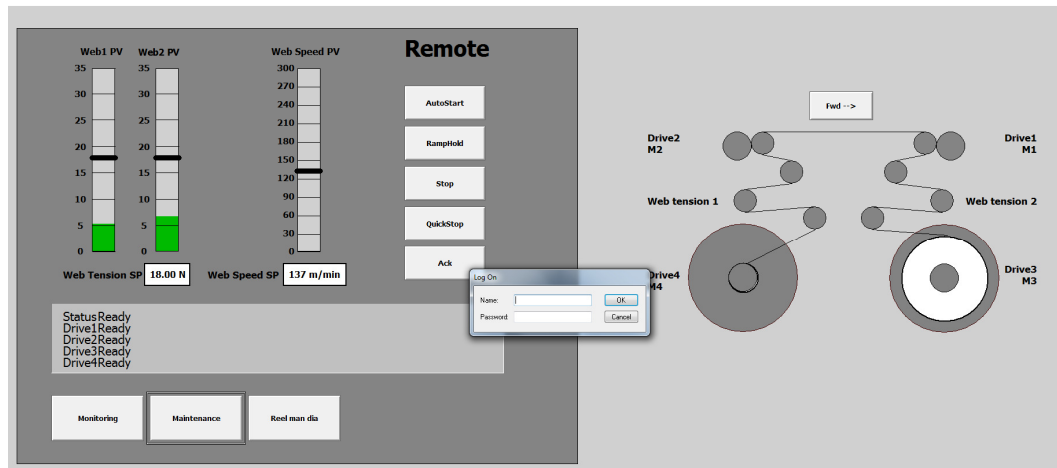
7.2 Käyttöliittymän ohjelmointi

Käyttöliittymän suunnittelussa lähtökohtana oli pitää ulkoasu mahdollisimman yksinkertaisena ja helppokäyttöisenä. Lisäksi käyttöliittymän tuli sisältää käyttöönottajien määrittelemiä ominaisuuksia. Näitä olivat nauhan kireyden ja nopeuden valvonnan ulkoasu, sekä niiden asetuksen ulkoasu. Liittymässä piti olla myös käynnistys, pysäytys, pikapysäytys ja ramphold-toiminto. Ramphold-toiminnolla pystyi jäädyttämään nopeusohjeen muutoksen kiihdytys- ja jarrutustilanteissa. Painikkeen vapautuksen jälkeen nopeuden muutos jatkui normaalisti.

Peruskäytösivun lisäksi tehtiin monitorointisivu ja eräänlainen huoltosivu, josta hallittiin erikoistoimintoja. Käyttöliittymän lopullinen ulkoasu muodostui käyttökokemusten ja palautteen perusteella.

7.2.1 Pääsivu

Pääsivulle tuli kaikki peruskäytön kannalta olennaiset ohjaukset. Näitä olivat käynnistys, ramphold, pysäytys, pikapysäytys, kelaussuunnan valinta, vikatietojen nollaus, kireysohjeen syöttö ja nopeusohjeen syöttö. Pääsivulta pystyi valvomaan nauhan kireyttä ja nopeutta. Lisäksi pääsivulla oli prosessiin liittyvä status-tieto, jolla voitiin indikoida missä tilassa tai missä vaiheessa kelaussekvenssiä laite oli. Taajuusmuuttajien tila- ja vikatiedot oli myös näkyvissä pääsivulla. Pääsivu on kuviossa 50.

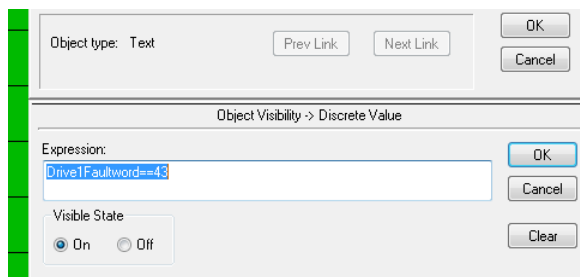


Kuvio 50 Käyttöliittymän pääsivu

Kireys- ja nopeusohjeen pystyi antamaan, joko kirjoittamalla asetuspiste kenttään tai käyttämällä liukusäädintä. Palkissa oleva liukusäädin osoitti asetuspisteen ja vihreä palkki indikoi mitattua prosessiarvoa.

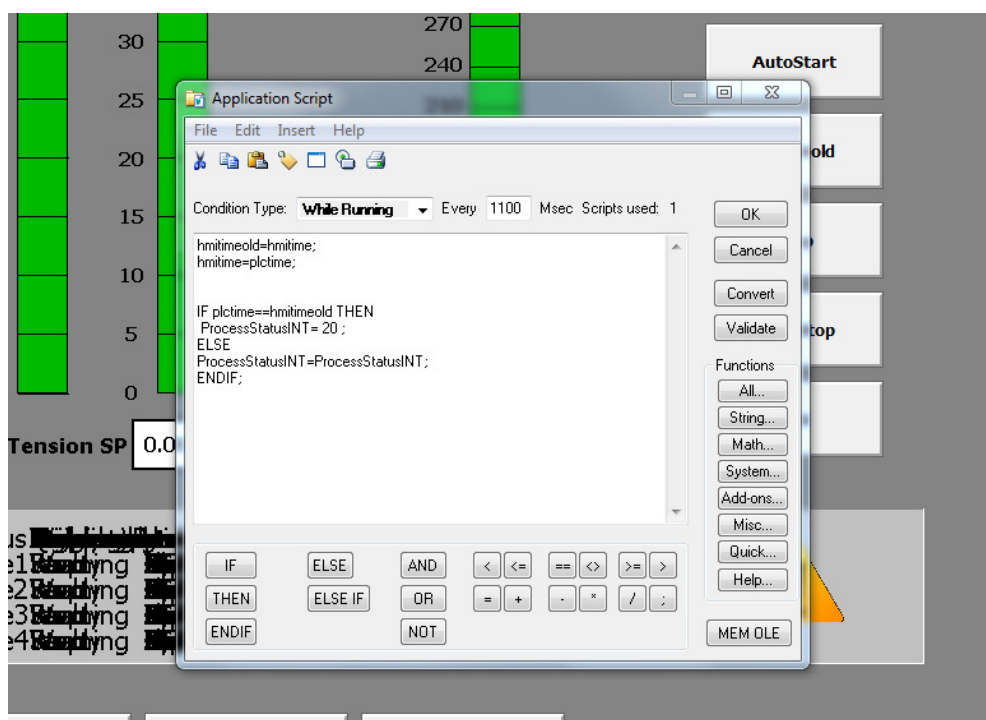
Pääsivun oikeassa laidassa oli hahmotelma laitteesta. Hahmotelman yläpuolella olevalla painikkeella sai valittua kelaussuunnan ja painikkeen teksti indikoi valitun suunnan. Painikkeen käyttö oli estetty kelauksen ollessa käynnissä. Hahmotelmassa olevat kelat ja niihin liittyvät viivat olivat dynaamisia objekteja. Hahmotelmassa olevien kelojen koko muuttui halkaisijan laskennan mukaan. Keloihin liittyvät viivat kiertyivät halkaisijan mukaan. Tällä tavoin niiden toinen pää oli aina samassa kohdassa ja toinen pää seurasi kelan ulkoreunaa.

Pääsivun alaosassa olevassa kentässä kerrottiin prosessin ja taajuusmuuttajien tilatietoja. Prosessin tilatiedot tulivat logiikalta integer-muuttujana. Muuttuja linkitettiin tilatietoteksteihin, jotka tulivat näkyviksi, kun muuttujan arvo vastasi kyseistä tietoa. Kuviossa 51 on esimerkki integer-muuttujan arvon liittämisestä tietyn objektin näkyvyyteen. Jokaisesta taajuusmuuttajasta tuli myös tilatieto ja vikatieto omana integer-muuttujana.



Kuvio 51 Vikatiedon linkitys integer-muuttuun

Pääsivulle haluttiin myös tieto logiikan ja käyttöliittymän välisen yhteyden katkeamisesta. Yhteyden tarkastamista varten tuotiin logiikalta sekunnin välein muuttuva integer-muuttuja. Window Makerilla tehtiin scripti, millä tarkastettiin integerin-muuttujan arvon muutos. Kuviossa 52 on tehty scripti. Scripti ajettiin 1,1 sekunnin välein, jotta vältettäisiin mahdolliset logiikan ja PC:n välisen väyläliikennöinnin viiveen aiheuttamat vikailmoitukset.

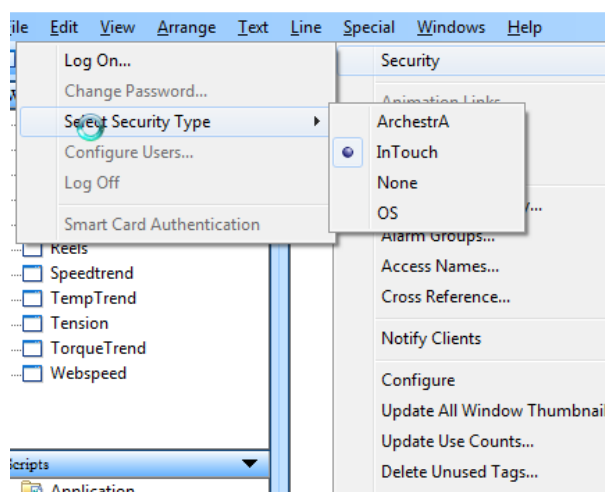


Kuvio 52 Yhteyden tarkistus scripti

Pääsivun alaosassa oli kolme painiketta, joilla aukesi lisäikkunoita. Monitoring-painikkeella avattiin valvontaikkuna, josta pystyi seuraamaan tarkemmin prosessiin ja toimilaitteisiin liittyviä tietoja. Maintenance-painikkeella aukesi ikkuna,

josta voitiin ohjata kelaimen erikoistoimintoja. Reel man dia-painikkeella aukesi ikkuna, josta kelan halkaisija voitiin asettaa käsin.

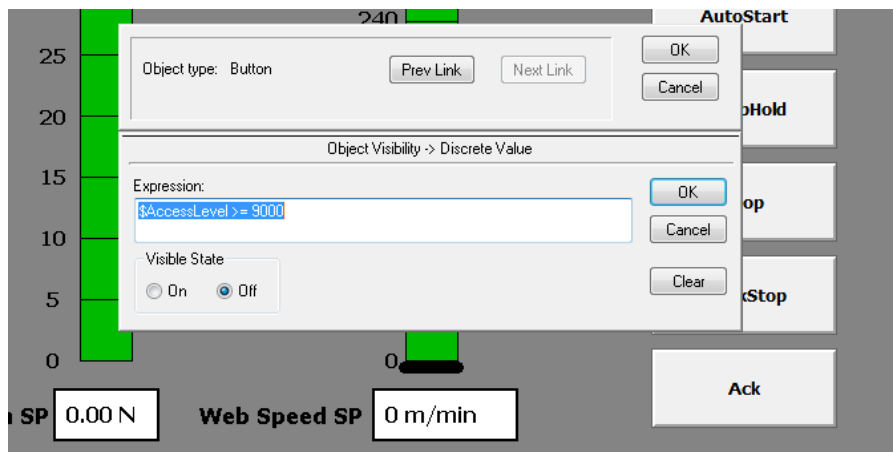
Maintenance-ikkuna haluttiin suojata salasanalla väärinkäytön estämiseksi. Salasanasuojaus toteutettiin InTouchin omalla security-toiminnolla, joka piti aktiivoida Window Makerin-valikosta. Kuviossa 53 on Window Makerin security-valikko.



Kuvio 53 Window Maker security-valikko

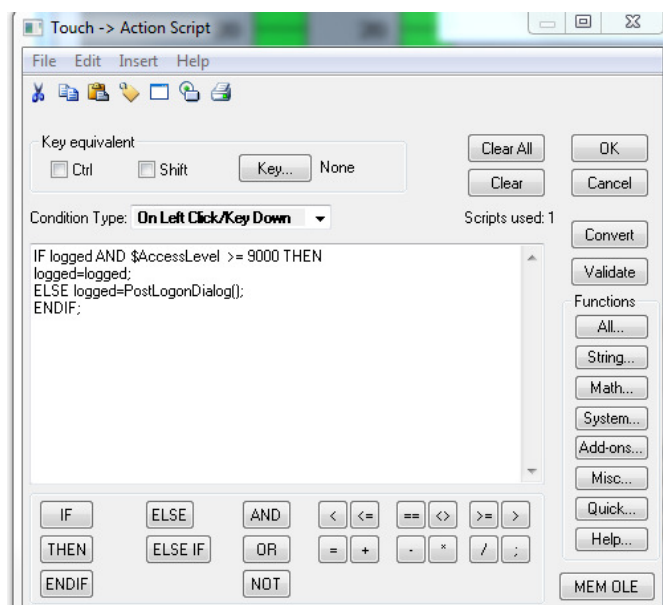
Securit -toiminnon aktivoinnin jälkeen ohjelma piti käynnistää uudelleen ja tämän jälkeen voitiin kirjautua sisään ohjelman oletus käyttäjätunnuksella ja salasanalla. Kirjautumisen jälkeen voitiin lisätä käyttäjiä. Lisätyille käyttäjille määriteltiin salasana ja acceslevel. Acceslevel on InTouchin sisäinen muuttuja millä määritellään käyttäjätunnuksen valtuudet. Acceslevel määriteltiin välillä 0-9999 ja käyttöliittymän objektien toiminnolle voitiin määritellä vaadittava Acceslevel.

Asetuksista määriteltiin myös automaattinen uloskirjautumisaika. Salasanalla suojaaminen toteutettiin tekemällä maintenance-painikkeen päälle samannäköinen painike. Painike oli näkyvissä, jos sisäänkirjautuneen käyttäjän Acceslevel oli alle 9000. Kuviossa 54 on esitetty painikkeen näkyvyysehto.



Kuvio 54 Acceslevel-toiminnon liittäminen objektiin

Painikkeeseen tehtiin lisäksi kuvion 55 mukainen scripti. Sillä avattiin sisäänkirjautumisikkuna, mikäli käyttäjätunnuksen Acceslevel oli liian alhainen.

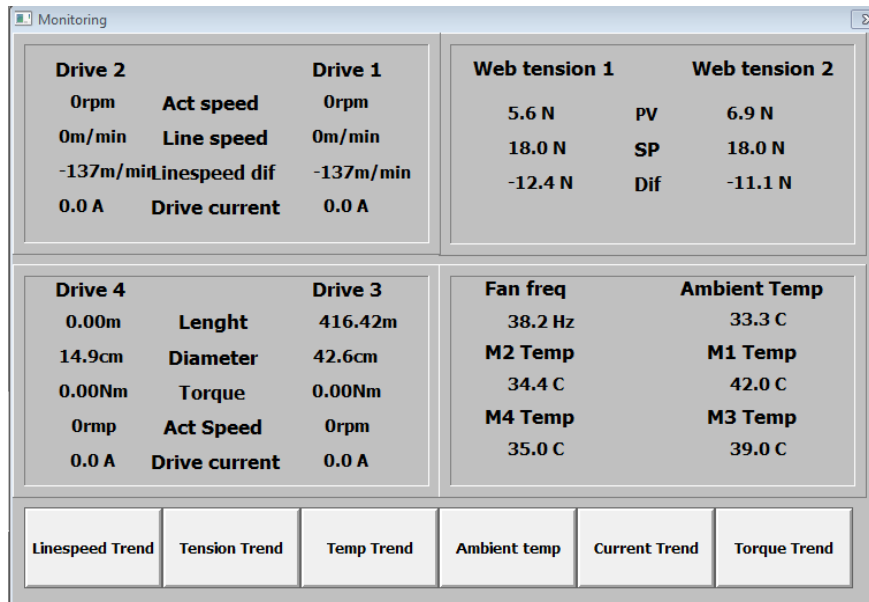


Kuvio 55 Sisäänkirjautumisikkunan avaus scripti

Onnistuneen sisäänkirjautumisen jälkeen ylimääräinen painike hävisi näkyvistä ja sen alla sijaitsevalla samannäköisellä painikkeella sai avattua maintenance-ikkunan. Automaattisen uloskirjautumisen jälkeen sisäänkirjautumisikkunan avaava painike tuli taas näkyviin.

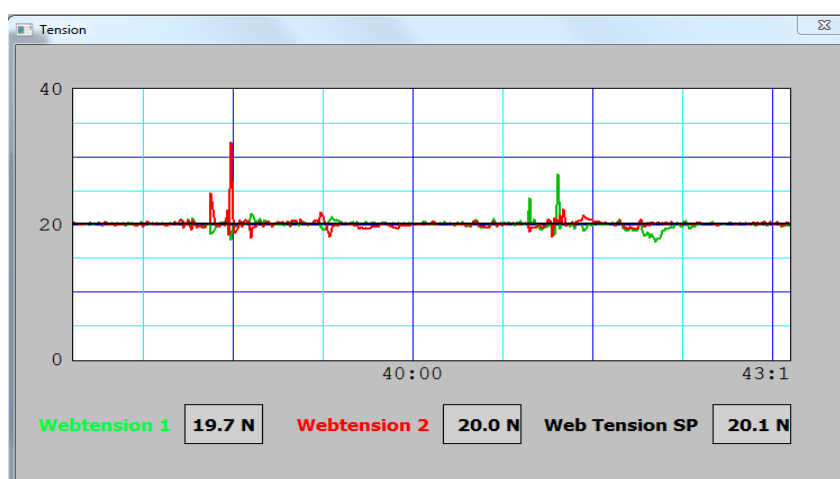
7.2.2 Monitorointi-ikkuna

Käyttöliittymään tehtiin myös erillinen ikkuna, josta voitiin valvoa taajuusmuuttajien- ja prosessin tietoja. Monitoroitavien tietojen määrää rajoitti käytössä olevien tagien määrä. Kuviossa 56 on monitorointi-ikkuna.



Kuvio 56 Monitorointi-ikkuna

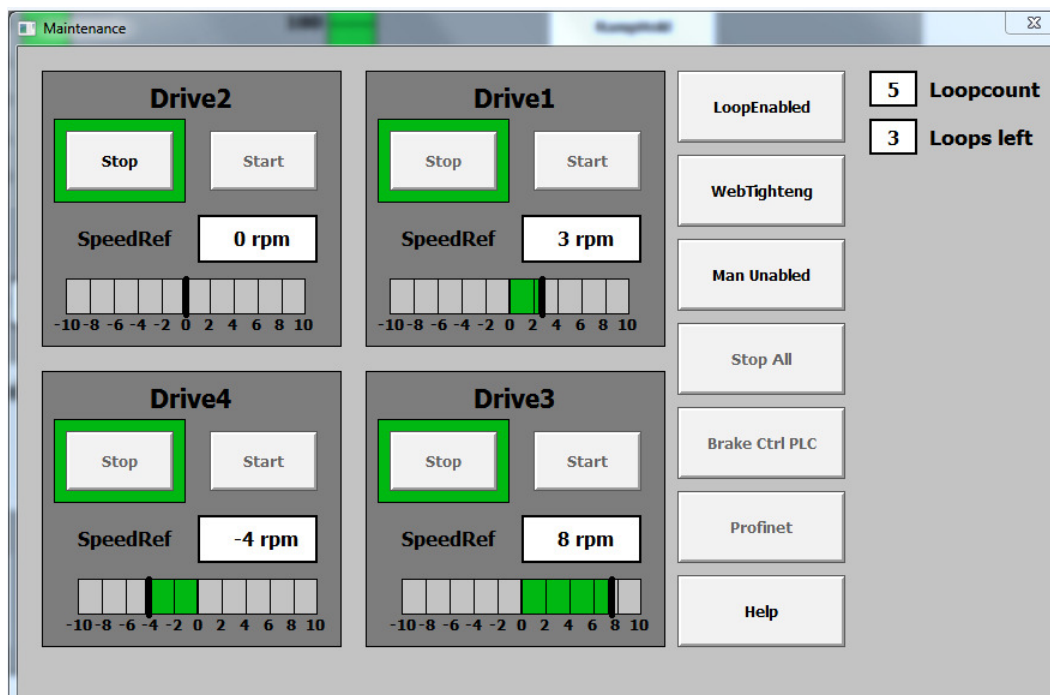
Monitorointi-ikkunan alaosassa olevista painikkeista sai avattua kuvion 57 kaltaisen historiaikkunan.



Kuvio 57 Historiaikkuna

7.2.3 Huoltoikkuna

Käyttöliittymään tehtiin myös kuvion 58 mukainen Maintenance-ikkuna, josta voitiin käyttää laitteen erikoistoimintoja.

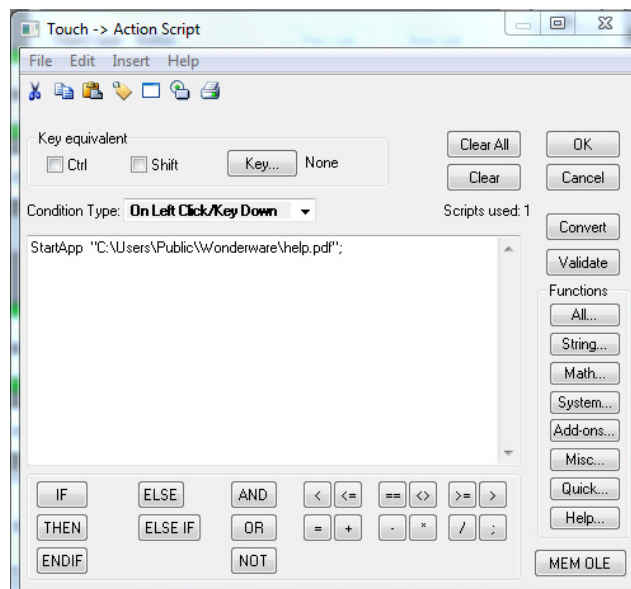


Kuvio 58 Maintenance-ikkuna

Huoltoikkunasta voitiin kytkeä Loop-toiminto päälle, asettaa sille haluttu loop-määrä ja tarkkailla niiden jäljellä olevaa määrää. Ikkunasta sai kytkettyä päälle myös erillisen kiristyssekvenssin, jolla saatiin kiristettyä nauha sen ollessa pahasti löysällä.

Huoltoikkunasta oli mahdollista ohjata yksittäistä taajuusmuuttajaa. Ensin piti asettaa Man-tila aktiiviseksi. Tämän jälkeen voitiin käynnistää yksittäiset taajuusmuuttajat ja ohjata niitä nopeusohjeella. Nopeusohjeen pystyi antamaan kirjoittamalla tai käyttämällä liukusäädintä. Nopeusohjeen rajoiksi asetettiin -10..10 rpm. Stop All-painikkeella sai pysäytettyä kaikki taajuusmuuttajat yhtäaikaaisesti, kun manuaaliohjaus oli käytössä.

Ikkunasta sai valittua myös jarrujen ohjauspaikan ja käytettävän kenttäväylän. Näiden käyttö oli sallittua vain, kun laite oli pysähtyneenä local-tilassa. Help-painikkeesta sai avattua laitteen käyttöohjeen. Painikkeen scripti on kuviossa 59.



Kuvio 59 Ohjeen avaus scripti

7.2.4 Window Viewer-asetukset

Ohjelmoitua käyttöliittymää käytettiin InTouchin Window Viewer-ohjelmalla. Kehitystilassa Window Viewerillä on mahdollista avata kaikki ikkunat ilman sisäänkirjautumista ja avata käyttöliittymäohjelma Window Makerilla. Window Makerin valikoista saa auki Window Viewerissä käytettävät asetukset. Valikosta asetettiin Window Viewer-ikkunan nimiökenttä ja valikot pois käytöstä. Ohjelma asetettiin myös toimimaan aina täydessä ikkunassa. Näillä asetuksilla näytöllä ei näy mitään muuta, kuin ohjelmoitu käyttöliittymä. Valikosta määriteltiin pääsivu koti-ikkunaksi, joka aukeaa ohjelman avauksen yhteydessä.

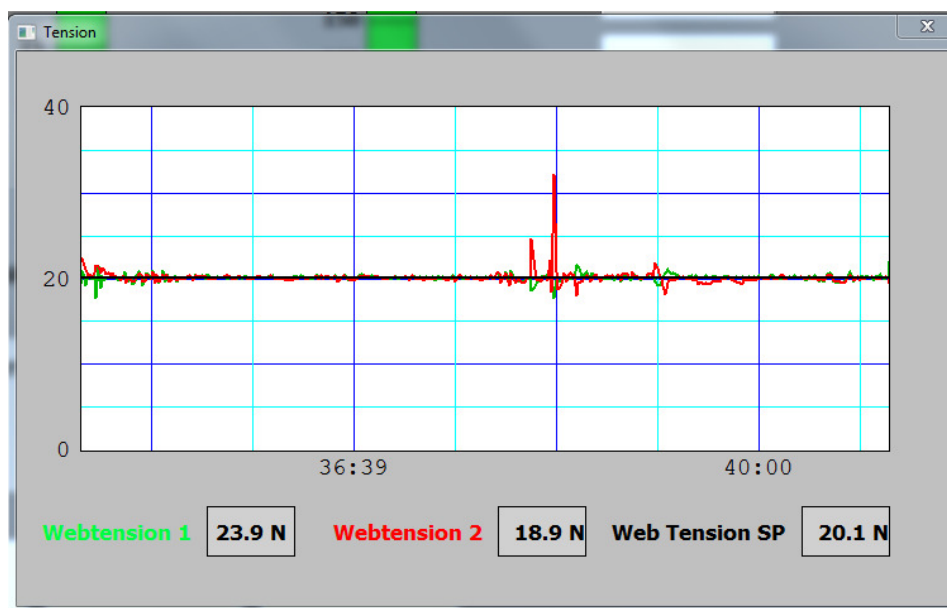
7.2.5 PC:n asetukset

PC:lle määriteltiin ajoitettuja tehtäviä, jotta käyttöliittymä saataisiin käyttövalmiiksi automaattisesti. SMC asetettiin aukeamaan aina, kun tietty käyttäjä kirjautuu tietokoneelle. SMC:ssä oli määritelty DASSIDirect3 toimimaan automatic server-toiminnolla. Näin SMC loi automaattisesti yhteyden logiikan ja käyttöliit-

tymän välille ohjelman auetessa. Ajoitettuihin tehtäviin lisättiin myös bat-tiedosto, jolla avattiin Window Viewer SMC:n auettua. Käynnistystä varten tehtiin bat - tiedosto, jotta voitaisiin lisätä mahdollisia lisätoimintoja tietokoneen käytön rajoittamista varten.

8 ESIMERKKIAJO

Kelaimen tärkeimpänä ominaisuutena oli nauhan kireyden säätö. Käyttöliittymään tuotiin nauhan kireysmittauksesta keskiarvo, jotta sen tulkinta olisi helpompaa. Kuviossa 60 on kireysmittauksen historiatieto, kun kelaimella on ajettu kaksi kierrosta.



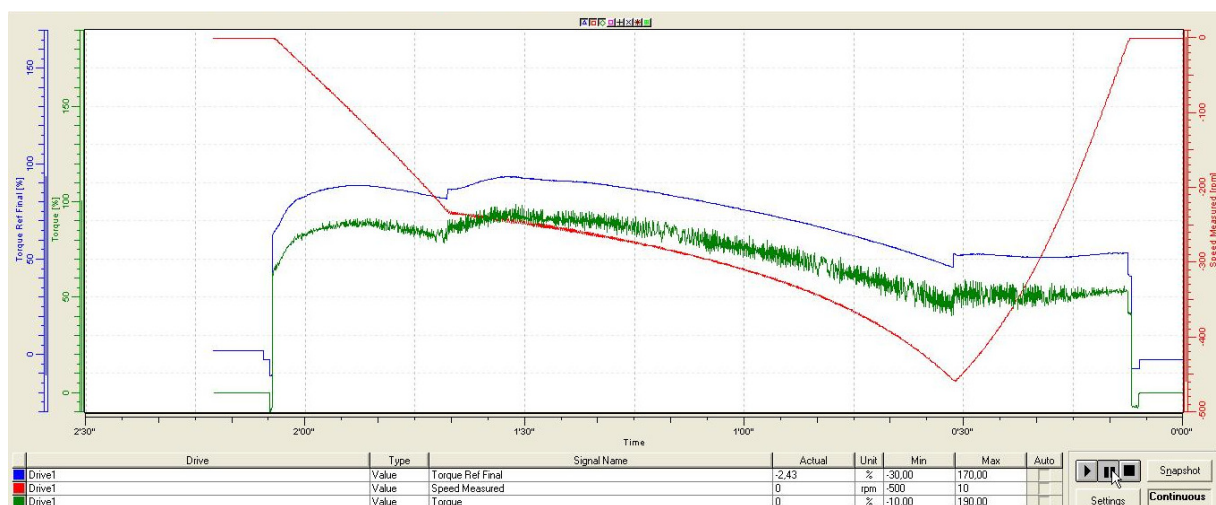
Kuvio 60 Kireyshistoria

Automaattisen kelausajan aikana nauhan kireys saadaan pidettyä hyvin asetuksessa. Kuviossa 60 näkyvät kireyden korkeammat arvot ovat tulleet kelaimen vaihdettua suuntaa.

Nauhan kireyden tarkkuudelle ei ollut työn tilauksen vaiheessa mitään määritelmää. Alkuperäinen vaatimus kelaimelle oli, että se kelausi nauhan päästä päähän automaattisesti ja siihen lisättäisiin toimintoja työn edistymisen mukaan.

Kireyssäädön parantaminen olisi vaatinut jo suurempia muutoksia ohjelmaan, ja mekaaniset osat olisi pitänyt saada täysin linjaan. Joissakin kohdissa kelainta rullat eivät olleet keskenään linjassa, jolloin matka kelojen reunojen välillä on erisuuri. Tämä aiheuttaa voiman epätasaisen jakaantumisen nauhan suhteen. Lasi-kuitunauha ei ole joustavaa materiaalia, mistä johtuen nauhaan kohdistuva epätasainen voima jakaantuu leveyssuunnassa pienelle alueelle. Tällaisessa tapauksessa

nauhan toinen reuna on kireä ja toinen reuna on löysä, mikä aiheuttaa nauhan liik-
kumista sivuttaissuunnassa.

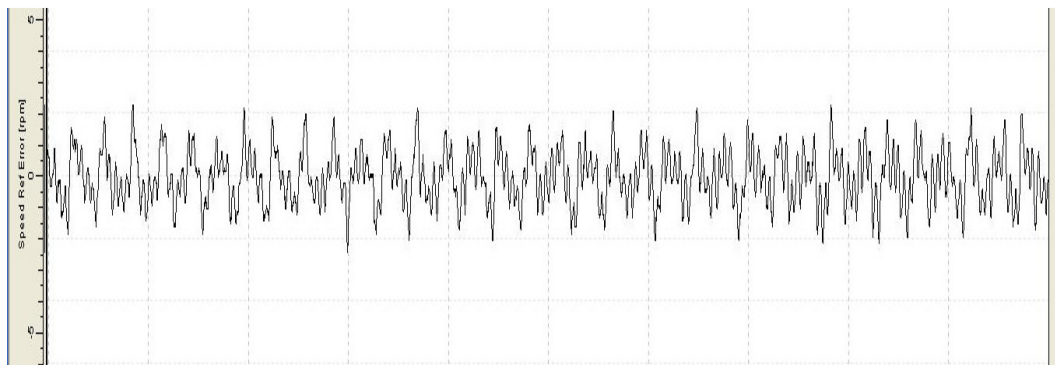


Kuvio 61 Aukikelaimen mittaukset

Kuviossa 61 on NCDrivellä otettu kuva aukikelaimen momenttiohjeesta, toteutuneesta momentista ja nopeudesta ajon ajalta. Kuviossa punainen väri on kelan pyörimisnopeus, sininen on momenttiohje ja vihreä on taajuusmuuttajan laskema toteutunut momentti. Momenttiohje ja toteutunut momentti on samassa skaalassa, mutta momenttiohjeen minimi- ja maksimiarvoa on muutettu, että ne eivät olisi päällekkäin.

Punaisella merkitystä nopeusmittauksesta voidaan helposti erottaa kiihdytysvaihe, tasainen nopeus ja hidastus. Momenttiohjeesta voidaan tarkastaa momenttiohjeen laskennan tarkkuutta. Momenttiohjeen tulisi muuttua mahdollisimman tasaisesti. Poikkeuksena kiihdytyksen loppu ja hidastuksen alku, joissa tulisi näkyä hitausmomentin kompensoinnin aiheuttama momenttiohjeen muutos.

Kuviossa 61 näkyy kiihdytysvaiheen loputtua momenttiohjeen askelmainen nousu. Hidastusvaiheen alettua ohjeeseen tulee myös askelmainen nousu. Kiihdytyksen alkuvaiheessa ja kiihdytyksen loputtua momenttiohjeessa näkyy aaltomaista liikettä. Ne ovat syntyneet, kun kireyssäätäjä on korjannut momenttiohjetta. Mikäli laskentaan ei tulisi lainkaan virhettä, momenttiohjeessa ei ilmenisi aaltoilua.

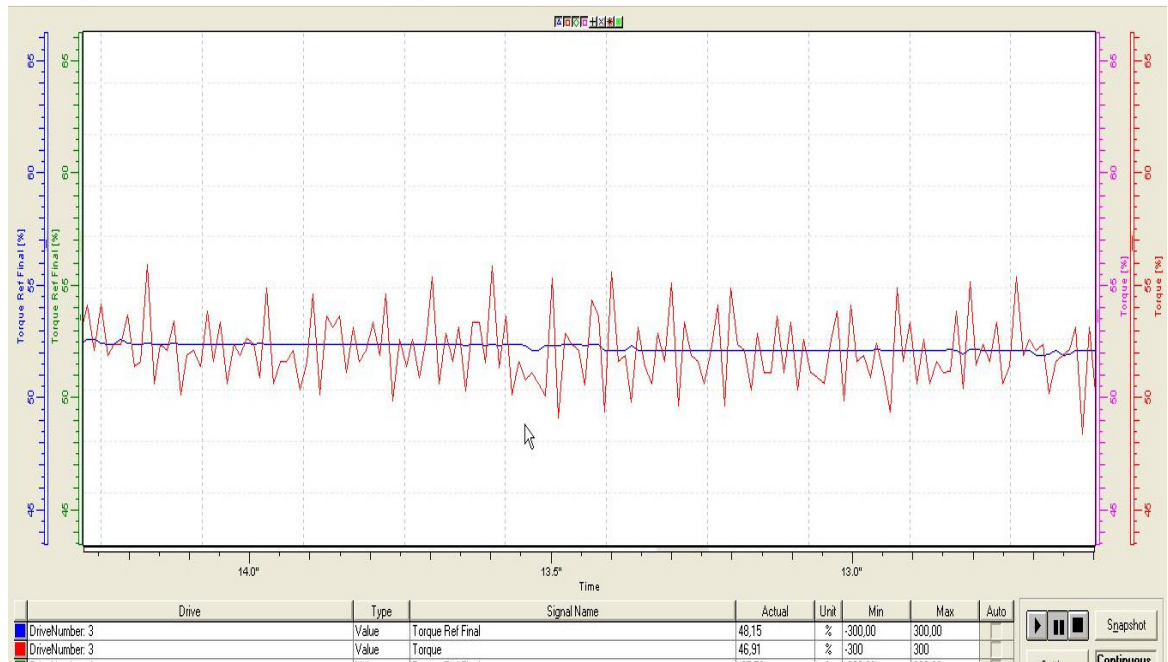


Kuvio 62 Nopeusvirhe

Momenttiohjeeseen vaikuttavien laskentojen tarkkuus perustuu täysin nopeusmittauksen tarkkuuteen. Nopeusmittauksen virheen ja nopeuden heilunnan vaikutus kertaantuu momenttiohjeessa.

Nopeusmittausten perusteella lasketaan kelojen halkaisijat. Halkaisijoista lasketaan momenttiohjeen perusosa ja kelan massa. Massasta lasketaan kitkan kompensointi. Halkaisijasta ja massasta lasketaan massahitausmomentti. Halkaisijan avulla suhteutetaan nopeusohjeen derivaatta kelan derivaataksi. Nopeusmittauksen virhe vaikuttaa siis kaikkiin momenttiohjeen muodostukseen liittyviin laskuihin.

Kuviossa 62 on nopeusohjeen ja mitatun nopeuden erotus käyttöönoton jälkeen. Mittaus on tehty CAN-väylän kautta, jolloin on saatu aikaiseksi 5 ms mittaussykli. Heittoa on ± 2 rpm suhteessa nopeusohjeeseen. Täydellä linjanopeudella heitto jää 1,6 promilleen. Ennen käyttöönottoa nopeusheitto oli noin ± 10 rpm.



Kuvio 63 Momenttiohje ja toteutunut momentti

Kuviossa 63 on momenttiohje ja toteutunut momentti, kun asetettu linjanopeus on saavutettu. Mittauksissa heittely on noin $\pm 2,5$ prosenttiyksikköä. Moottorin nimellismomentin ollessa 11,2 Nm tämä tarkoittaa $\pm 0,28$ Nm heittelyä ja voidaan olettaa kelaimen suuren massahitausmomentin vaimentavan momentin heiluntaa.

9 KÄYTÖN RAJOITUKSET

Kelaindemo on tarkoitettu koulutuskäyttöön. Tämän vuoksi koekäyttöjen avulla on pyritty löytämään mahdollisimman paljon virhekäyttömahdollisuuksia. Esille tulleet virhekäyttömahdollisuudet on estetty ohjelmallisesti. Kelaimen sovellus ja käyttöliittymä on pyritty ohjelmoimaan niin, että kuka tahansa voi käyttää sitä ilman hajottamisen vaaraa. Tämä oli ohjelmoinnin kannalta haastavin osuus. Laitteessa oli useita sekvenssejä ja tietyt toiminnot olivat käytettävissä vain tietyillä sekvensseillä. Sekvenssien käynnistämiseksi oli useita ehtoja ja niiden keskeyttämisille oli tiettyjä ehtoja, jotka riippuivat sekvenssin vaiheesta. Lisäksi kaikki toiminnot piti saada palautettua samaan tilanteeseen yhdellä painikkeella virheen jälkeen. Tällä tavoin kuitenkin saatiin minimoitua väärinkäytön mahdollisuudet.

Laitteelle määriteltiin ohjelmallisesti rajat kireys- ja nopeusohjeelle. Nopeusohjeelle annettiin vain yläraja. Ylärajaksi määriteltiin 300 m/min, jolloin linjan moottoreiden lähtötaajuus on noin 60 Hz. Kelojen moottoreiden pyörimisnopeudet ovat suurempien halkaisijoiden vuoksi pienempiä ja niiden nopeus ei nouse missään vaiheessa yli nimellisen nopeuden. Kireysohjeiden rajaksi tuli 10–50 N. Yläraja määräytyi koeajojen perusteella. Moottoreiden virtaraja tuli vastaan suuremmalla kireydellä ja alaraja määräytyi kireyssäädön tarkkuuden vuoksi. Alle 10 N kireydellä moottoreiden momenttiohje oli hyvin pieni ja taajuusmuuttajien oli hankala pitää pieni momentti tasaisena. Momentin heilunnan takia nauha pääsi liikkumaan sivuttaissuunnassa ja siihen tuli ajoittain löysiä kohtia.

10 YHTEENVETO JA POHDINTA

Opinnäytetyö oli mielestäni hyvin laaja. Se sisälsi automaatio suunnittelun kannalta lähes kaikkia vaiheet. Työ alkoi toiminnallisuuden suunnittelusta, joka toteutettiin yhdessä työn ensimmäisen vaiheen toteuttaneen Jani Oravasaaren kanssa. Hardware -asetukset vaativat myös enemmän työtä kahden väylän takia. Työtä varten piti tutustua tarkemmin väyläliikennöinnin asetuksiin ja niiden syvempiin tarkoituksiin.

Ohjelmointiympäristönä oli TIA-Portal, josta oli hieman aikaisempaa kokemusta. Osittain uusi ohjelmointiympäristö aiheutti muutamia ongelmia ohjelmointitilanteissa, mutta lopulta TIA-Portal osoittautui Manageria toimivammaksi ohjelmointiympäristöksi. Käytettävä kirjasto sisälsi vain perus ohjelmointilohkot, minkä vuoksi rampit ja keskiarvolaskentalohkotkin piti ohjelmoida itse.

Logiikkaohjelman jälkeen piti luoda käyttöliittymä PC:n ja logiikan välille yhteys. PC:lle piti myös ohjelmoida käyttöliittymä. Ohjelman ja käyttöliittymän valmistuttua laitetta koeajettiin ja ohjelmaan haluttiin monia rajoituksia ja lisäyksiä. Työssä piti tehdä myös monia asioita, jotka eivät liittyneet logiikan ohjelmointiin, kuten taajuusmuuttajan parametointia ja mekaanisten osien säätöä.

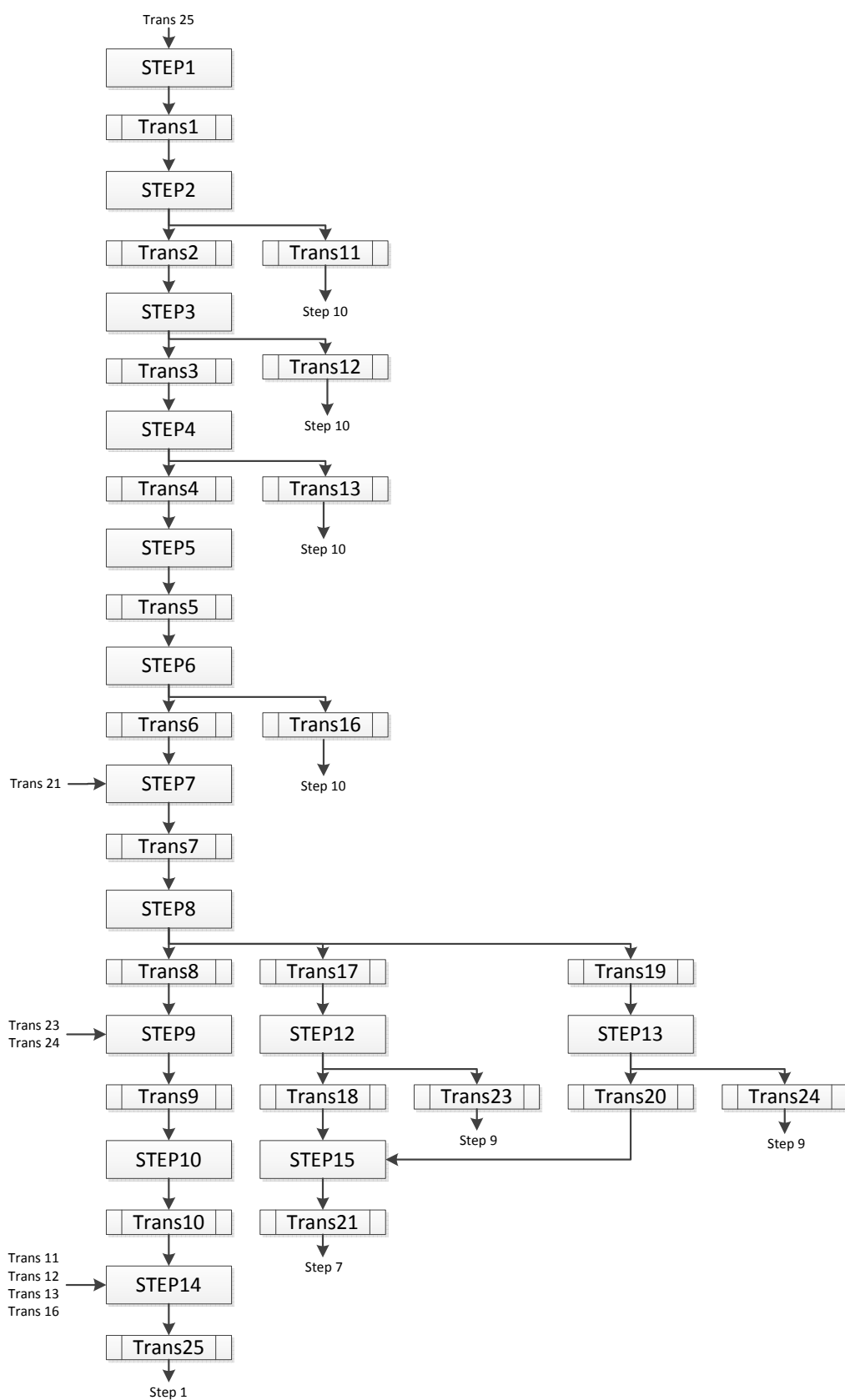
Opinnäytetyö oli erittäin mielenkiintoinen juuri sen monipuolisuuden takia. Työssä sai toteuttaa kaikki työn vaiheet, eikä vain yksittäistä osaa. Työntekoon annettiin vapaat kädet. Tämä oli hyvä, kun sai itse määritellä ohjelman sisällön ja toteutuksen. Toisaalta vapaus hankaloitti suunnittelua, kun ei aina ollut selkeää tietoa halutun lopputuloksen yksityiskohdista.

LÄHTEET

/1/ Siemens TIA-Portal. Viitattu 26.4.2014. http://www.siemens.fi/fi/industry/teollisuuden_tuotteet_ja_ratkaisut/tuotesivut/tia_portal.php

/2/ APFIF40 Vacon system interface application II 'SIA II' application manual

/3/ Tammertekniikka 2010. Tekniikan kaavasto 8.painos.



Askel	Ohjaukset	Siirtymävaihtoehdot
Step1	Ei ohjauksia	Trans1
Step2	Taajuusmuuttajien ohjauslohko auto tilaan Taajuusmuuttajat 3 ja 4 nopeusohjaukselle Resetoidaan Käynnistyskäsky	Trans2, Trans11
Step3	Käynnistää taajuusmuuttajat	Trans3, Trans12
Step4	Nauhan kiristys	Trans4, Trans13
Step5	Nolla nopeusohje	Trans5
Step6	Taajuusmuuttajat 3 ja 4 momenttiohjaukselle	Trans6,Trans16
Step7	Rampaa nopeutta kohti asetusta	Trans7
Step8	Nopeus nollaan	Trans8, Trans17,Trans19
Step9	Ramppi pois päältä Jarrut päälle jos jarrujen PLC ohjaus aktiivinen	Trans9
Step10	Taajuusmuuttajat stop tilaan Stop ja Quickstop käskyn resetointi	Trans10
Step12	Suunnanvalinta bitti pois päältä Loop laskuriin lisätään yksi	Trans18,Trans23
Step13	Suunnanvalinta bitti pois päältä Loop laskuriin lisätään yksi	Trans20,Trans24
Step14	Siirretään vanha nopeusohje takaisin	Trans25
Step15	Siirretään vanha nopeusohje takaisin	Trans21

Siirtymä	Siirtymä ehdot	Siirtymäkohde
Trans1	Taajuusmuuttajat seis Ei muita aktiivisia askelia Taajuusmuuttajat Ready tilassa Käynnistyskäsky käyttöliittymästä	Step2
Trans2	Taajuusmuuttajien lohko autotilassa Taajuusmuuttajat 3 ja 4 nopeusohjauksella	Step3
Trans3	Taajuusmuuttajat käynnissä	Step4
Trans4	Nauha kireänä molemmin puolin	Step5
Trans5	Moottoreiden nopeus on nolla	Step6
Trans6	Taajuusmuuttajat 3 ja 4 momenttiohjauksella Nauhan kireys asetusarvon hystereesin sisällä	Step7
Trans7	Aukikelaimen nauha <= nauhan ennakointi Stop, Quickstop, fault tai warning Moottori tai ympäristö yllilämpö Oviraja Local tilan valinta	Step8
Trans8	Moottoreiden nopeus on nolla Loop tila ei käytössä	Step9
Trans9	Ramppi on pois päältä Jarrut on päällä jos jarrujen PLC control on aktiivinen	
Trans10	Taajuusmuuttajat stop tilassa	Step10
Trans11	Stop, Quickstop, fault tai warning	Step10
Trans12	Stop, Quickstop, fault tai warning	Step10
Trans13	Stop, Quickstop, fault tai warning	Step10
Trans16	Stop, Quickstop fault tai warning	Step10
Trans17	Moottoreiden nopeus on nolla Loop tila on käytössä Suunnanvalintabitti on tosi	Step12
Trans18	Suunnanvalinta bitti on epätosi ja loop tila aktiivinen	Step15
Trans19	Moottoreiden nopeus on nolla Loop tila on käytössä Suunnanvalintabitti on epätosi	Step13
Trans20	Suunnanvalinta bitti on tosi ja loop tila aktiivinen	Step15
Trans21	Uusi nopeusohje = vanha	Step7
Trans23	Loop laskuri = asetettu loop määrä	Step9
Trans24	Loop laskuri = asetettu loop määrä	Step9
Trans25	Uusi nopeusohje = vanha	Step1